

Universidad Carlos III de Madrid

Escuela Politécnica Superior



Grado en Ingeniería Electrónica Industrial y Automática

Trabajo Fin de Grado

**IMPLEMENTACIÓN DE
APLICACIÓN ANDROID PARA
CONTROL DE DISPOSITIVO
ASISTENCIAL PRESSMATIC**

Autor: Javier Ladrón de Guevara Ojalvo

Tutor: Edwin Daniel Oña Simbaña

**IMPLEMENTACIÓN DE APLICACIÓN ANDROID PARA CONTROL DE DISPOSITIVO ASISTENCIAL
PRESSMATIC**





Agradecimientos

En primer lugar, agradecer a mi tutor y al director del presente trabajo por darme la oportunidad de desarrollar esta tarea que se ajustaba a mis necesidades y objetivos de entre los trabajos propuestos (fue el que me resultó más interesante).

También a mi familia por sus sacrificios durante todos estos años de esfuerzo y cambio, pues sin su ayuda no hubiese sido posible haber completado esta primera fase de trabajo.

A profesores, antiguos compañeros y amigos, por su apoyo y sus esfuerzos conmigo.



Resumen

Pressmatic es un dispositivo asistencial para personas con discapacidad manual formado por dos componentes separados: cuerpo principal, que dispone de una pantalla táctil con el interfaz de usuario, y los cabezales intercambiables, que realizan tareas de corte o agarre según la herramienta.

Según el instrumento elegido es posible realizar una tarea, pues encajan en el cuerpo base del aparato y se accionan según el sistema de transmisión a través de un enganche homogéneo.

El usuario puede controlar la herramienta de dos formas: mediante la pantalla táctil del dispositivo o a través de la aplicación con un terminal con sistema operativo Android. Este control se basa en el envío de órdenes desde el terminal al dispositivo Pressmatic utilizando la tecnología bluetooth.

Este trabajo se centra en el desarrollo de la segunda versión de la aplicación Android, mejorando tanto su funcionalidad como la interfaz gráfica, la cual ha sido diseñada en un trabajo previo, teniendo en cuenta la experiencia de usuario. Además, se han añadido funcionalidades como el reconocimiento de voz, la capacidad multilenguaje y asegurando una conexión robusta durante el uso de la aplicación.

Palabras clave: Pressmatic, Android, aplicación, reconocimiento, voz, asistencial, control y bluetooth.



Abstract

Pressmatic is a device that pretends assist people with manual handicaps to realize actions, as cut or attach, through electromechanical pulses.

The object possesses a transmission system which can control several instruments using either user interface, on its tactile screen, or an Android application on a smartphone. This tasks perform the role played by the thumb and forefinger.

Depending on the tool selected by the user it is possible to choose the frequency of acting. Also, controlling start and stop and other implementations.

This bachelor thesis focuses on the development of the second version of the Android application, improving both its functionality and the graphical interface, which has been designed in a previous work, taking into account the user experience.

In addition, features such as voice recognition, multi-language capability and ensuring a robust connection have been added during application use.

Keywords: Pressmatic, Android, application, recognition, voice, assist, control and bluetooth.

**IMPLEMENTACIÓN DE APLICACIÓN ANDROID PARA CONTROL DE DISPOSITIVO ASISTENCIAL
PRESSMATIC**





Índice General

| | |
|--|-----------|
| Agradecimientos..... | 2 |
| Resumen..... | 3 |
| Abstract..... | 4 |
| Capítulo 1 | 12 |
| Introducción | 13 |
| 1.1 Descripción del problema | 15 |
| 1.2 Motivación | 16 |
| 1.3 Objetivos del trabajo | 17 |
| 1.4 Estructura del documento | 18 |
| Capítulo 2 | 19 |
| Estado del Arte | 21 |
| Capítulo 3 | 31 |
| Descripción del sistema..... | 32 |
| 3.1 Introducción | 32 |
| 3.2 Análisis del sistema | 32 |
| 3.2.1 Descripción de las características funcionales | 37 |
| 3.2.2 Restricciones del sistema..... | 39 |
| 3.2.3 Entorno operacional | 40 |
| 3.2.4 Especificación de casos de uso..... | 41 |
| 3.2.5 Especificación de requisitos | 52 |
| 3.2.5.1 Requisitos funcionales | 52 |
| 3.2.5.2 Requisitos no funcionales | 55 |
| 3.3 Diseño del sistema | 57 |



| | |
|---|------------|
| 3.3.1 Arquitectura del sistema..... | 59 |
| 3.3.2 Descripción general del sistema..... | 67 |
| 3.3.3 Descripción de componentes..... | 70 |
| 3.3.3.1 Reconocimiento de voz..... | 70 |
| 3.3.3.2 Diseño de la interfaz | 71 |
| 3.3.3.3 Modificación del lenguaje..... | 72 |
| 3.3.3.4 Órdenes enviadas mediante Bluetooth | 73 |
| Capítulo 4 | 74 |
| Experimentación..... | 75 |
| 4.1 Fase 1: Simulaciones..... | 76 |
| 4.2 Fase 2: Pruebas con el prototipo real | 77 |
| Capítulo 5 | 88 |
| Gestión del proyecto..... | 89 |
| 5.1 Marco regulador | 90 |
| 5.2 Entorno socio-económico | 95 |
| 5.3 Descripción de las fases del proyecto..... | 98 |
| Capítulo 6 | 100 |
| Conclusiones y trabajos futuros | 101 |
| 6.1 Conclusiones generales | 101 |
| 6.2 Conclusiones referentes a los objetivos | 102 |
| 6.3 Trabajos futuros | 103 |
| Capítulo 7 | 109 |
| Anexos..... | 110 |
| 7.1 Manual de instalación..... | 110 |
| 7.2 Manual de usuario..... | 112 |



Índice de figuras

| | |
|--|----|
| Figura 1 – Tipos de robots..... | 14 |
| Figura 2 – Robot asistencial Asibot..... | 15 |
| Figura 3 – Áreas de la robótica..... | 19 |
| Figura 4 – Robots humanoides para asistencia e interacción..... | 21 |
| Figura 5 -Robots asistenciales..... | 22 |
| Figura 6 – Versiones de Android..... | 24 |
| Figura 7 – Control de aparatos domésticos mediante bluetooth..... | 28 |
| Figura 8 – Control por voz de una silla de ruedas inteligente..... | 28 |
| Figura 9 – Dispositivo Pressmatic y aplicación Android original..... | 31 |
| Figura 10 – Electrónica de control Pressmatic..... | 32 |
| Figura 11 – Herramientas Pressmatic..... | 33 |
| Figura 12 – Comparación aplicaciones para Pressmatic..... | 34 |
| Figura 13 – Diagrama de flujo..... | 36 |
| Figura 14 – Especificación Instrucciones..... | 39 |
| Figura 15 – Especificaciones conexión bluetooth..... | 40 |
| Figura 16 – Especificaciones herramientas y órdenes a Pressmatic..... | 42 |
| Figura 17 – Comunicación bluetooth..... | 56 |
| Figura 18 – Arquitectura de la plataforma..... | 57 |
| Figura 19 – Arquitectura de la plataforma: Aplicaciones..... | 58 |
| Figura 20 – Arquitectura de la plataforma: Armazón de aplicaciones | 58 |
| Figura 21 – Arquitectura de la plataforma: Librerías..... | 59 |
| Figura 22 – Arquitectura de la plataforma: Android Runtime..... | 60 |
| Figura 23 – Arquitectura de la plataforma: Kernel de Linux..... | 60 |



| | |
|--|-----|
| Figura 24 –Diagrama de conexión bluetooth..... | 62 |
| Figura 25 – Diagrama reconocimiento de voz Android..... | 63 |
| Figura 26 – Logo “Habla ahora” de Google..... | 64 |
| Figura 27 – Instrucciones de Pressmatic..... | 65 |
| Figura 28 – Actividades bluetooth. Izquierda: habilitar. Derecha conectar..... | 66 |
| Figura 29 – Herramientas disponibles..... | 67 |
| Figura 30 – Pantalla de inicio..... | 70 |
| Figura 31 – Progreso del diseño aplicación Pressmatic..... | 86 |
| Figura 32 – Fases de un proyecto..... | 93 |
| Figura 33 – Diagrama de Gantt..... | 96 |
| Figura 34 – Logo Pressmatic..... | 103 |
| Figura 35 – Manual de instalación..... | 108 |
| Figura 36 – Manual de instalación 2..... | 109 |
| Figura 37 – Manual de instrucciones 1..... | 110 |
| Figura 38 – Manual de instrucciones 2..... | 111 |
| Figura 39 – Manual de instrucciones 3..... | 112 |
| Figura 40 – Manual de instrucciones 4..... | 113 |
| Figura 41 – Manual de instrucciones 5..... | 114 |
| Figura 42 – Manual de instrucciones 6..... | 115 |



Índice de tablas

| | |
|---|----|
| Tabla 1 - Casos de uso | 44 |
| Tabla 2 – CU-001 | 45 |
| Tabla 3 – CU-002 | 45 |
| Tabla 4 – CU-003 | 46 |
| Tabla 5 – CU-004 | 46 |
| Tabla 6 – CU-005 | 47 |
| Tabla 7 – CU-006 | 47 |
| Tabla 8 – CU-007 | 48 |
| Tabla 9 – CU-008 | 48 |
| Tabla 10 – Tabla de requisitos | 50 |
| Tabla 11 – RF-001 | 50 |
| Tabla 12 - RF-002 | 51 |
| Tabla 13 - RF-003 | 51 |
| Tabla 14 - RF-004 | 51 |
| Tabla 15 - RF-004 | 52 |
| Tabla 16 - RF-005 | 52 |
| Tabla 17 - RF-006 | 52 |
| Tabla 18- RF-007 | 53 |
| Tabla 19 - RNF-001 | 53 |
| Tabla 20 – RNF-002 | 53 |
| Tabla 21 – RNF-003 | 54 |
| Tabla 22 – Organización del trabajo | 94 |
| Tabla 23– Precio recursos | 98 |





Capítulo 1



Introducción

Desde hace siglos, el ser humano pretende construir máquinas para imitar partes de su cuerpo, su estructura y movimiento. En Egipto, unieron brazos mecánicos a las estatuas y; en Grecia, fueron contruidos los primeros sistemas hidráulicos en los templos.

En la antigüedad, se denominaban autómatas y no robots, ya que la palabra robot apareció en el siglo XVIII, cuando Joseph Jacquard inventó una máquina textil programable a través de tarjetas perforadas. Más tarde, junto a la Revolución Industrial, fueron desarrollados más agentes mecánicos.

La palabra robot se utilizó en la obra "Los Robots Universales de Rossum", de Karel Capek, una historia de un hombre que fabrica un robot que le asesina. La palabra checa 'Robota' (trabajado forzado) fue traducida en el término robot.

En 1939, Isaac Asimov, estableció el término Robótica y las conocidas como "Tres Leyes de Robótica". Así, este término podría definirse de la siguiente forma:

La robótica es una disciplina que estudia las ramas de diseño y construcción de máquinas para realizar tareas humanas de forma inteligente. Forma parte de la ingeniería mecatrónica, eléctrica, electrónica e informática y se pretende que ocupe parte de la sociedad laboral y cotidiana con el paso del tiempo. [12]

Hoy día, está compuesta por varias áreas, según: la morfología, procedimiento de control o aplicación a desarrollar. Estas áreas se pueden identificar (brevemente) como:

- robótica industrial: automatización de procesos industriales.
- robótica educativa: genera entornos de aprendizaje.
- robótica médica: compatibiliza el cirujano con el robot para mejorar los procedimientos quirúrgicos.
- robótica asistencial: presta soporte a personas con impedimentos (tema de este trabajo).



En la siguiente figura (figura 1) se puede ver una pequeña descripción de los distintos tipos de robots.



Robot Teleoperado

-Sin cerebro, solo cuerpo

Robot Automático

- Repite movimientos
- Comportamiento reactivo
- Ambiente estructurado
(adaptado para el robot)

Robot Autónomo

- Toma decisiones propias
- Aprende y se adapta
- Ambientes no estructurados

Figura 1: Tipos de robots.

La robótica asistencial se especializa en la implementación y el diseño de equipos que interactúan directamente con la persona para su rehabilitación, ya sea por la pérdida física de algún miembro o el déficit de capacidad en la movilidad.

Tradicionalmente, en la rehabilitación se sustituía el miembro disminuido facilitando al discapacitado una prótesis controlada de forma muscular mediante impulsos eléctricos (mioeléctrica). De esta forma, se permitía a individuos cuadripléjicos manejarse en su entorno mejorando su locomoción y proviniendo cierta autonomía para sus actividades cotidianas.



Debido a los fallos producidos por estos sistemas y a lo invasivo de estas terapias, se ha avanzado en sistemas de ayuda a la rehabilitación, que no necesitan de inclusión en el cuerpo del paciente y aun así mejoran notablemente su calidad de vida. Véase figura 2:



Figura 2: Robot asistencial Asibot.

1.1 Descripción del problema

En las últimas décadas se han realizado muchos estudios acerca del uso de robots para asistir a seres humanos y llevar a cabo una labor de ayuda en su rehabilitación. El crecimiento de estos sistemas ha sido necesario debido al envejecimiento de la población, las guerras y la búsqueda de soluciones a la pérdida de miembros por enfermedades.

El campo de la robótica de la asistencia a humanos incluye diversos ingenios mecánicos: miembros artificiales, robots de soporte a terapias de rehabilitación o robots para proveer asistencia personal en hospitales.



Los sistemas robóticos asistencias suelen presentar una compleja interfaz para controlarlos, lo que dificulta su utilización sin una formación previa o la presencia de un asistente.

Mediante el uso de sistemas de control a través de terminales inteligentes, se pretende dar acceso a una gran parte de la población, tanto en el sentido económico como en el de eficiencia, pudiendo ser utilizados en cualquier situación.

1.2 Motivación

Con el objetivo de mejorar la inclusión de la robótica en el día a día, es necesario ser capaces de combinar su control y mantenimiento con las herramientas que utilizamos usualmente. Una de las más utilizadas, sin duda, son los dispositivos portables inteligentes (tales como teléfonos móviles, tabletas...); si se consigue utilizarlos tanto para simplificar el uso de robots como para mejorar la vida cotidiana, ya se ha hecho un gran avance.

Para facilitar el manejo de las máquinas de asistencia a humanos y adaptarlo a la sociedad actual, se suele combinar la interfaz de usuario con una aplicación que se instala en estos terminales (haciéndola dinámica y atractiva).

De esta manera, es posible un control del aparato de manera flexible y sencilla, además de adaptarlo a las situaciones cotidianas de uso junto a familiares o amigos que no tengan ningún conocimiento tecnológico.



1.3 Objetivos del trabajo

En base a ensayos de usabilidad con usuarios reales, se validó la fiabilidad técnica y la aceptación por parte del usuario de Pressmatic. Además, se destacó la utilidad del control del dispositivo por medio de un Smartphone. Teniendo en cuenta la experiencia de usuario, una nueva interfaz gráfica fue diseñada, mucho más funcional y atractiva que la implementada para el primer prototipo. [1]

El objetivo principal de este trabajo es desarrollar una aplicación Android para controlar el dispositivo asistencial Pressmatic a partir de una ya existente. El diseño gráfico ha sido diseñado en un trabajo previo, teniendo en cuenta los ensayos de usabilidad realizados con usuarios reales.

Además, se mejorará la funcionalidad implementada en la aplicación inicial, destacando la inclusión de un sistema de reconocimiento de voz para el control del dispositivo asistencial Pressmatic. Esta implementación debe mantener la capacidad de conectar con el dispositivo y mantener el enlace establecido.

A continuación, se detallan estos puntos, exigencias para la resolución de este proyecto:

- Aplicación multilenguaje: capaz de mostrar una interfaz en español e inglés cuando el usuario lo deseara.
- Aplicación multimodal: con una herramienta para poder utilizarla mediante el habla para minimizar el control táctil.
- Diseño atractivo con experiencia al usuario, desarrollo y mejora de funcionalidades.



1.4 Estructura del documento

Esta memoria se divide en 7 capítulos, los cuales están formados por epígrafes para facilitar la lectura del texto y la organización de los temas. Todos incluyen imágenes relacionadas con los temas tratados para clarificar el texto (también tablas).

El temario ha sido seleccionado según la relación de los elementos del proyecto con las ciencias y estudios actuales, teniendo en cuenta su finalidad y uso, además de las herramientas y actores necesarios para su desarrollo.

Al final del sexto capítulo se encuentra el apartado bibliográfico con las fuentes de consultas del presente trabajo. A continuación, se detalla la estructura de los capítulos:

- **Capítulo 1:** en este capítulo se realiza una introducción a la historia de la robótica, se exponen las cuestiones en materia de robótica asistencial y las motivaciones y objetivos con los que se pretende dar solución.
- **Capítulo 2:** en este capítulo se trata el estado del arte del trabajo, que detalla los comienzos de la robótica, su evolución cronológica, clasificación y uso. También, se realiza una introducción al sistema operativo para dispositivos electrónicos Android.
- **Capítulo 3:** en este capítulo se indica qué es el dispositivo Pressmatic, con qué fin fue desarrollado y qué objetivos han dado paso a la implementación de este trabajo. Se describe la arquitectura del Sistema Operativo Android, se lleva a cabo un análisis de la aplicación desarrollada, detallando el diagrama y los casos de uso, los requisitos y las funcionalidades de la aplicación.
- **Capítulo 4:** en este capítulo se detallan las pruebas que se han llevado a cabo para comprobar el correcto funcionamiento de la aplicación, desde el testado en el emulador hasta las pruebas con el dispositivo real.



- **Capítulo 5:** en este capítulo están incluidos el marco regulador vigente y el entorno socio-económico del proyecto realizado. Además, se describe la planificación que se ha seguido para desarrollar este Trabajo Fin de Grado y se detalla el presupuesto de su desarrollo.
- **Capítulo 6:** en este capítulo se aclaran las conclusiones obtenidas tras la realización del trabajo y los proyectos que pueden llevarse a cabo mejorando la funcionalidad. Además, se incluye un apartado de bibliografía donde se encuentran libros y páginas web consultados.
- **Capítulo 7:** en este capítulo se incluyen los anexos y manuales de instalación y ayuda al usuario, pretendiendo aclarar la forma en que ha de ser utilizada la aplicación, los requisitos de uso del reconocedor de voz y las instrucciones para adquirir el entorno de desarrollo.



Capítulo 2



Estado del Arte

Los seres humanos siempre han sentido curiosidad por máquinas y dispositivos capaces de simular los movimientos de los seres vivos. En Grecia, denominaron estas herramientas como “automatos” (palabra de la que deriva *autómata*), que se utiliza para calificar una máquina que imita la forma y las tareas que desempeña un ser animado.

El autómata más conocido a lo largo de la historia es el Gallo de Estrasburgo (que formaba parte del reloj de la catedral), que al dar la hora debía mover las alas y el pico. Desde entonces se ha trabajado en la evolución de estos mecanismos para su uso y disponibilidad al alcance de la mayoría [20].

La robótica continuó desarrollándose desde los años cuarenta gracias a dos tecnologías: el control numérico, pretendía dominar una máquina-herramienta mediante el uso de tarjetas perforadas; y la telequerica, que empleaba un ser humano, denominado tele operador, para transmitir sus movimientos en una posición distinta a través de un manipulador remoto.

Hoy en día se pueden clasificar los robots según su morfología, procedimiento de control y sus aplicaciones o tareas capaz de realizar. Desde la mejora de la producción industrial hasta la ayuda en terapias con enfermos y ancianos, su utilidad en los diversos campos sociales es cada vez mayor. Esto puede observarse en la siguiente figura (figura 3):



Figura 3: Áreas de la robótica



Gracias a los avances en el campo de la robótica se han introducido tecnologías en valoración clínica, rehabilitación y procesos terapéuticos capaces de superar tradicionales tratamientos de curación y quirúrgicos reduciendo el impacto en los pacientes y el tiempo de recuperación.

La robótica médica y la robótica asistencial (consideradas avanzadas), van más allá de los robots industriales con movimientos repetitivos. Son capaces tanto de aprender de su entorno como de conversar e interactuar con las personas sin dañarlas.

La mejora en la precisión de los movimientos, reducción de la fatiga del cirujano y las intervenciones a distancia han sido los principales progresos.

El primer robot conocido que dio lugar al comienzo de las investigaciones en este campo fue *Robodoc*, una máquina electromecánica diseñada por el ejército americano que posibilitaba la atención a distancia por medio de manipulación remota en situaciones de guerra.

En 1993 fue presentado Esopo, un robot esclavo con capacidad de reconocer comandos por voz. Aunque, la primera intervención quirúrgica transatlántica se dio con la aparición de Zeus y el comienzo de la cirugía por tele presencia.

Las aplicaciones de los robots en cirugía permiten realizar operaciones menos invasivas y en un tiempo menor, proporcionando al paciente un menor tiempo de recuperación. Se pueden clasificar de forma técnica como:

- Cirugía guiada por imagen: proporcionan una visualización en tres dimensiones. También se denomina laparoscopia robotizada, como ejemplo en la actualidad se presenta el sistema digital Da Vinci, un robot quirúrgico presentado en 2003 y desarrollado por *Intuitive Surgical Inc.*
- Cirugía mínimamente invasiva: debido al control en las desviaciones de la trayectoria y en la velocidad de los movimientos.



El principal inconveniente de estas tecnologías es su elevado coste (de entre 2-3 millones de dólares), tanto la inversión inicial como el mantenimiento y formación del personal sanitario.

El porcentaje de personas afectadas por discapacidades físicas, sensoriales o psíquicas asciende al 15 por ciento de la población mundial, según la *Organización Mundial de la Salud*. Estas discapacidades afectan a su integración social y laboral.

Los robots de asistencia están dedicados a proveer servicios a los seres humanos o a otros equipamientos de manera autónoma o semiautónoma. Aquellos que operan con humanos influyen en medicina, seguridad y entretenimiento; mientras que los de servicio a otros dispositivos se dedican a labores de mantenimiento, reparaciones, limpieza...

Así, a lo largo de la historia se han desarrollado soluciones y dispositivos tecnológicos para la asistencia a personas con discapacidades y para ayudar a recuperar funciones físicas deterioradas (solventar defectos motrices, amputaciones, lesiones...).

En la siguiente figura (figura 4) se observan dos robots humanoides capaces de relacionarse con seres vivos sin ocasionar peligro:

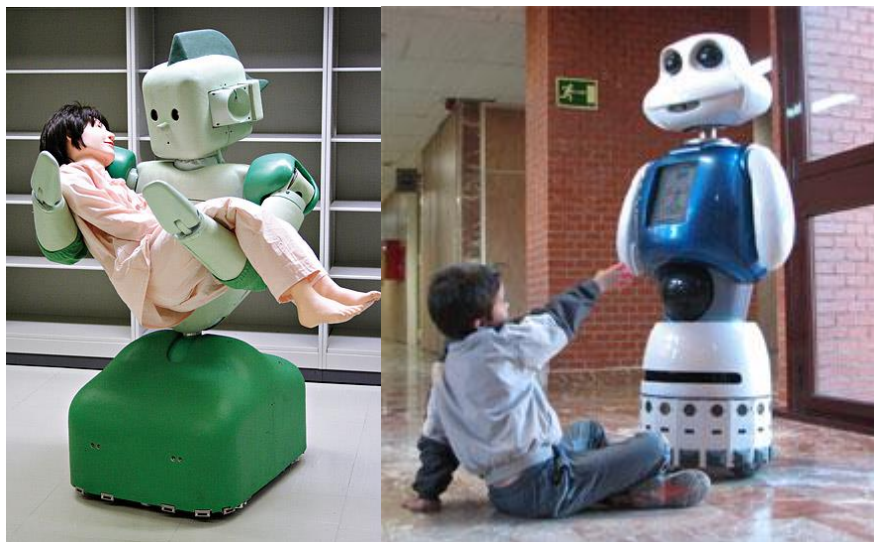


Figura 4: Robots humanoides para asistencia e interacción.



La *tecnología asistencial* (TA) persigue realzar la independencia de una persona con discapacidad funcional para realizar actividades diarias. Estos dispositivos pueden ser de apoyo: como sillas de ruedas, prótesis o audífonos; y dispositivos de ayuda visual, equipos y programas informáticos especializados, etcétera [13].

En la siguiente imagen (figura 5), se puede ver la variedad asistencial que existe en la actualidad dentro de la robótica. En diferentes situaciones y necesidades, se es capaz de dar respuesta a las carencias específicas de cada paciente:



Figura 5: Robots asistenciales.



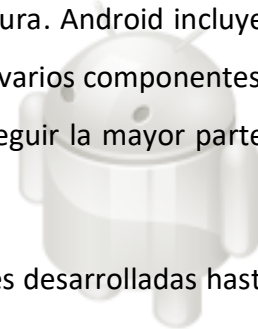
Para el control de estos dispositivos es necesario diseñar sistemas de manejo sencillo y eficaz para poder ser utilizados por cualquier usuario, adaptados a los sistemas actuales y de bajo coste o gratuitos. Este software de control puede ser desarrollado bajo distintas plataformas, como pueden ser Java, Arduino o Android entre otros.

Este proyecto se desarrolla sobre la plataforma Android (basada en el sistema operativo Linux) para dispositivos móviles. Es una implementación en código abierto y gratuito (sin pago de licencias).

Android fue desarrollado por el consorcio de compañías *Open Handset Alliance* como una pila de software para dispositivos móviles inteligentes que incluye: un Sistema Operativo, Middleware y aplicaciones de base. Se pueden crear aplicaciones mediante el SDK (*software development kit*) que provee la plataforma. Se basa en el lenguaje de programación Java y desde la versión 5.0 se ejecuta en ART (*Android Runtime*) como un nuevo entorno de ejecución [16]. Este entorno se caracteriza por:

- Compilar el Java “bytecode” durante la instalación de una aplicación en lugar del antiguo entorno “Dalvik”, una máquina virtual personalizada que se ejecuta en la parte superior de un núcleo de Linux, y que lo hacía al inicio de cada aplicación.
- Es uno de los “sistemas operativos” (en lenguaje coloquial de telefonía) más extendidos por el mundo, por lo que su utilización en proyectos y desarrollos posibilita una gran perspectiva global. Permite ejecutar procesos en segundo plano, soporta gráficos en 2 y 3 dimensiones y ofrece una interfaz de usuario muy potente y dinámica.
- La reutilización de componentes es la base de su arquitectura. Android incluye un conjunto de bibliotecas en el lenguaje C/C++ utilizadas por varios componentes del sistema y un set de bibliotecas base que utiliza para conseguir la mayor parte de las bibliotecas base de Java.

En la siguiente imagen (figura 6) se pueden ver las versiones desarrolladas hasta el momento:



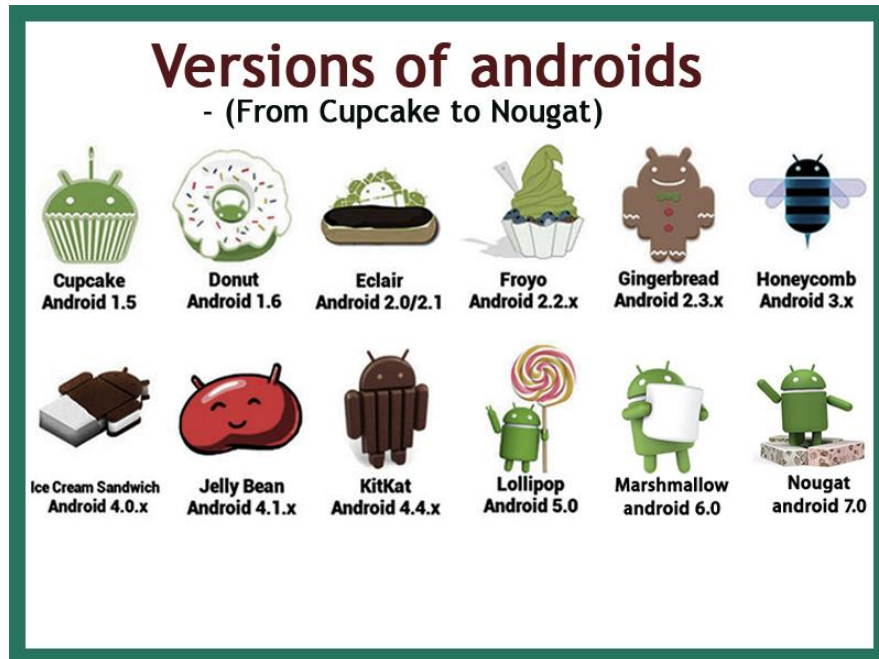


Figura 6: Versiones de Android.

Hoy día, ya son 14 las versiones que ha desarrollado Android para sus consumidores (contando con los prototipos), desde la primera versión lanzada en 2007 como prueba para su sistema hasta la que utilizan hoy los terminales más modernos.

Cada versión de Android ha incluido depuración de errores, mejoras en el diseño o aumento de funcionalidades, de esta manera se ha convertido en un sistema operativo que no queda obsoleto y que es capaz de competir con líderes como IOS de Apple [15].

A continuación, se va a realizar una breve descripción de cada una de las versiones desarrolladas por la compañía (sin tener en cuenta las versiones de prueba Alfa y Beta). Para ello, se va a utilizar la siguiente nomenclatura:

Nombre (versión, fecha de lanzamiento, nivel de API)

- Apple Pie (1.0, 23/09/2008, 1): primera versión comercial, permite navegar vía web, utilizar la cámara y descargar aplicaciones (en otras funciones).
- Banana Bread (1.1, 09/02/2009, 2): permite almacenar archivos adjuntos, mejora la visualización en llamadas y búsquedas en mapa y añade soporte al sistema.

- Cupcake (1.5, 27/04/2009, 3): incluye soporte a la interfaz de usuario, reproduce y graba sonidos y añade protocolos de intercambio de información.
- Donut (1.6, 15/09/2009, 4): se comienza a desarrollar el cuadro de búsqueda rápida, el Android Market y las versiones para diferentes terminales.
- Éclair (2.0–2.1, 26/10/2009, 5-7): se implementan los fondos animados, la navegación en tiempo real y la información sobre el tráfico.
- Froyo (2.2–2.2.3, 20/05/2010, 8): aumentan las funcionalidades, como el reconocimiento de voz, las zonas Wi-Fi y la velocidad de los dispositivos.
- Gingerbread (2.3–2.3.7, 06/12/2010, 9–10): sigue mejorando la velocidad para la experiencia del usuario, aumenta la duración de la batería y la compatibilidad NFC mejora la calidad de aplicaciones y juegos.
- Honeycomb (3.0–3.2.6, 22/02/2011, 11–13): inaugura la llegada de las tablets y la nueva interfaz de usuario, con más tamaño y potencia.
- Ice Cream Sandwich (4.0–4.0.4, 18 /10/2011, 14–15): permite personalizar el diseño de la interfaz, modificar la cantidad de datos utilizada e intercambiar contenidos.
- Jelly Bean (4.1–4.3.1, 09/07/2012, 16–18): continúa la mejora del diseño, Google avanza en el desarrollo de funcionalidades y se permite a un usuario utilizar varias cuentas.
- KitKat (4.4–4.4.4, 4.4W–4.4W.2, 31/10/2013, 19–20): Google termina de desarrollar su sintetizador de voz “Ok Google” y permite navegar por el terminal.
- Lollipop (5.0–5.1.1, 12/11/2014, 21–22): el diseño se adapta a todo tipo de dispositivos, además se hace personalizable y dinámico.
- Marshmallow (6.0–6.0.1, 05/10/2015, 23): las aplicaciones se vuelven inteligentes, la batería vuelve a aumentar su capacidad y el usuario es capaz de controlar su información.
- Nougat (7.0 - 7.1, 22/08/2016, 24 – 25): es posible modificar las notificaciones, mejora en el uso de la batería e incorpora la opción de multiventana de forma nativa.



En función de cada versión, es posible utilizar funcionalidades como reconocimientos de voz o táctil para implementar aplicaciones capaces de controlar otros dispositivos.

Hoy día, existen diversas aplicaciones para el control de dispositivos asistenciales o médicos que se presentan para sustituir complejos sistemas (software o hardware) de control para robots [13].

Android permite el control de cualquier otro sistema a través del acelerómetro del terminal, el bluetooth, wi-fi o usb; además, sus diversas funcionalidades habilitan estas aplicaciones maestro-esclavo a personas con discapacidad que encuentren en el interfaz táctil restricciones para su uso.

Las aplicaciones para dispositivos inteligentes con fines médicos o de asistencia se pueden dividir en 4 categorías, aunque se pueden combinar varias funcionalidades:

1. Cuidado y monitorización del paciente: tensión, niveles de glucosa, recuperación tras una intervención....
2. Aplicaciones de salud y deporte: tanto para personas de la tercera edad como jóvenes con o sin hándicaps motores.
3. Comunicación (entre sanitario y paciente, como la telediagnosís), educación sanitaria y búsqueda de información.
4. De referencia al estudio médico: para estudiantes del hámbito sanitario.

Además de la interfaz táctil, el desarrollador puede otorgar a la aplicación capacidades para mejorar la calidad de su usabilidad dependiendo del entorno y la situación del individuo. Estas competencias han incrementado la independencia y movilidad de las personas con discapacidad. Pueden clasificarse en:

- Control por reconocimiento de voz
- Control por gestos y reconocimiento de objetos y colores
- Control mediante el movimiento del terminal
- Control mediante señales o impulsos eléctricos



Algunas de las aplicaciones Android hoy disponibles que se utilizan para asistir a personas con discapacidad se pueden descargar y utilizar de forma complementaria al resto de aplicaciones presentes en el teléfono, lo que consigue mejorar la experiencia con los dispositivos a personas con deficiencias.

Estas aplicaciones permiten a personas con discapacidad visual, auditiva, motora o de algunas funciones (manual, habla...) utilizar los terminales con menos inconvenientes y controlar sus propios dispositivos asistenciales, lo que les otorga más independencia y mejora su calidad de vida. Como ejemplos se pueden exponer:

- Talkback: habilita el dispositivo para comunicarse mediante habla o vibraciones al ejecutar otras aplicaciones o añadir eventos.
- Voice Access: permite controlar el teléfono con la voz.
- BrailleBack: permite conectar una pantalla en braille mediante bluetooth al dispositivo para interactuar y escribir texto en braille en el terminal Android.
- Switch Access: permite interactuar con el dispositivo Android a través de switches, por lo que no es necesario utilizar la interfaz del terminal.
- Captions: captura las palabras que están siendo escuchadas por el terminal, convirtiéndolas en texto.
- Magnification Gestures: permite magnificar la pantalla (zoom).
- Large Text: modifica el tamaño del texto a lo más largo posible.
- High Contrast Text: modifica el contraste del texto haciéndolo más claro u oscuro.

Estas aplicaciones pueden considerarse también capacidades o funcionalidades a incluir en otros sistemas de control para mejorar y facilitar su utilización.

Así, es posible combinar una aplicación de reconocimiento de voz, transcripción de órdenes o reconocimiento de objetos con un software para controlar un robot u dispositivo, lo que permitiría al usuario que éste evitase objetos, reconociera formas, entendiera órdenes habladas o sea utilizado para guiar visualmente a una persona.

Un ejemplo de aplicaciones capaces de controlar mediante bluetooth u otros protocolos a diferentes dispositivos y operar de manera asistencial para mejorar la calidad de vida diaria de individuos con funcionalidades reducidas son las siguientes:

- Figura 7: aplicación Android para controlar dispositivos domésticos por bluetooth mediante comandos de voz [6].

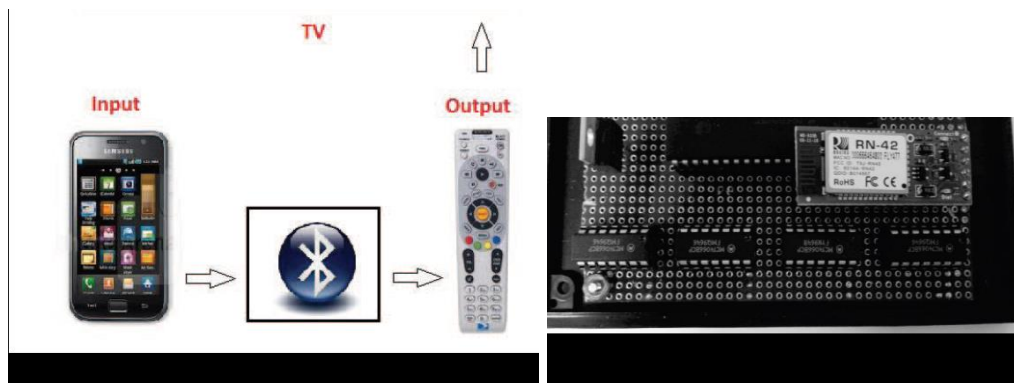


Figura 7: Control de aparatos domésticos mediante bluetooth.

- Figura 8: aplicación Android para controlar una silla de ruedas inteligente por bluetooth mediante reconocimiento de voz y posiciones del terminal [7].

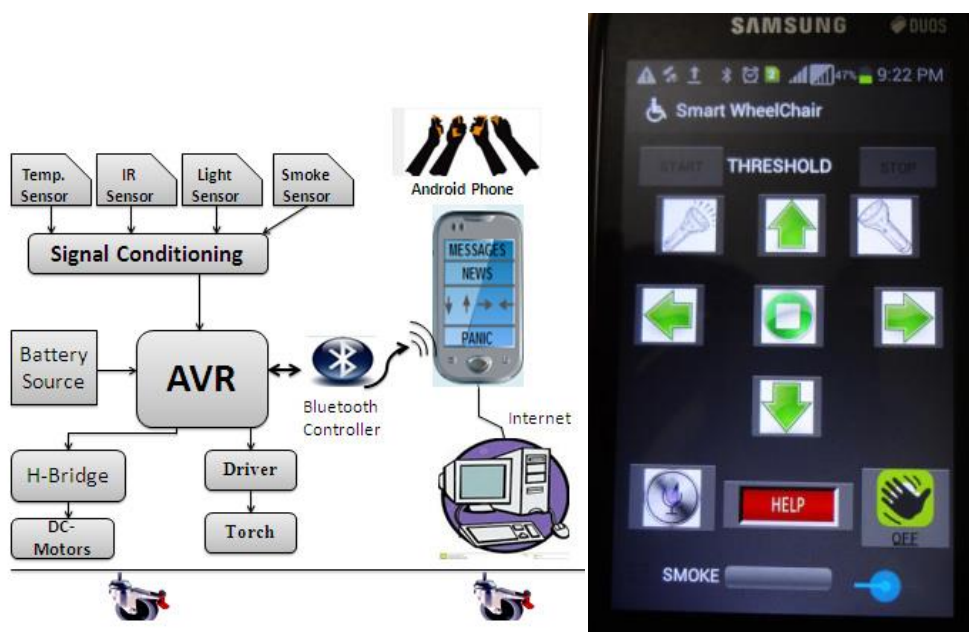


Figura 8: Control por voz de una silla de ruedas inteligente.



Capítulo 3



Descripción del sistema

3.1 Introducción

En la actualidad, existen varios estudios que demuestran la utilidad de las aplicaciones móviles para influir en la recuperación de pacientes en rehabilitación o contribuir a facilitar sus acciones cotidianas.

En este proyecto se pretende utilizar la tecnología para que permita asistir a personas con discapacidad, en este caso, destreza manual reducida o hándicaps motores en las manos. Esto se consigue mediante la implementación de una aplicación que facilita el control de un dispositivo asistencial, y que extiende su usabilidad al incorporar herramientas como el reconocimiento de voz, permitiendo ampliar el caso de uso a usuarios con afectaciones neuromusculares más severas.

Pressmatic es un proyecto cuya patente se basó en el diseño de los investigadores del *RoboticsLab*, que buscaban una herramienta para asistir a pacientes con movilidad manual reducida en tareas complejas (como cortar o agarrar objetos) [14]. El dispositivo consta de dos elementos diferenciados:

1. El cuerpo principal, que contiene el sistema de transmisión y una pantalla táctil.
2. Los cabezales, herramientas de uso: tijeras, cortaúñas y pinzas.

Pressmatic generará movimientos automáticos pre-programados, adecuados a la herramienta seleccionada. Este control orientado a la herramienta busca que el modo de empleo sea más sencillo e intuitivo, que requiera de mínimo esfuerzo físico. [21]

Es necesario sostener el cuerpo principal y manipular la interfaz de usuario al mismo tiempo para realizar las tareas. Se elige una de las herramientas y el correspondiente programa mediante la pantalla táctil del aparato o la aplicación Android.



3.2 Análisis del sistema

Pressmatic es un aparato electromecánico formado por un subsistema electrónico de control que se opera a través de la interfaz de usuario. Dada su morfología y peso, es posible manipular el instrumento con una sola mano mientras que con la otra se pueden intercambiar los cabezales o manipular la aplicación.

Los exámenes de ejecución probaron la utilidad de la herramienta para ayudar a personas con alguna minusvalía en las manos (o con pérdida de capacidad de movimiento) para realizar acciones como agarrar objetos pequeños o abrir bolsas de comida precintada. [2]. En la siguiente figura (figura 9) se observa el proyecto base de este trabajo:



Figura 9: Dispositivo Pressmatic y aplicación Android original.

La aplicación se ha desarrollado para gobernar al dispositivo mediante el envío de comandos (a través de sockets que utilizan el protocolo bluetooth) que se encuentran previamente programados en lenguaje Arduino en Pressmatic.

Al ejecutar una orden se producen ciclos automáticos de apertura o cierre sobre el cabezal que simulan los esfuerzos de los dedos pulgar e índice. Además, se puede modificar la velocidad de estos ciclos desde la aplicación.

En la siguiente figura (figura 10) se puede apreciar el interior del cuerpo del dispositivo, con la electrónica de control y los actuadores que permiten la transformación de los pulsos eléctricos en pequeños movimientos en la herramienta elegida.

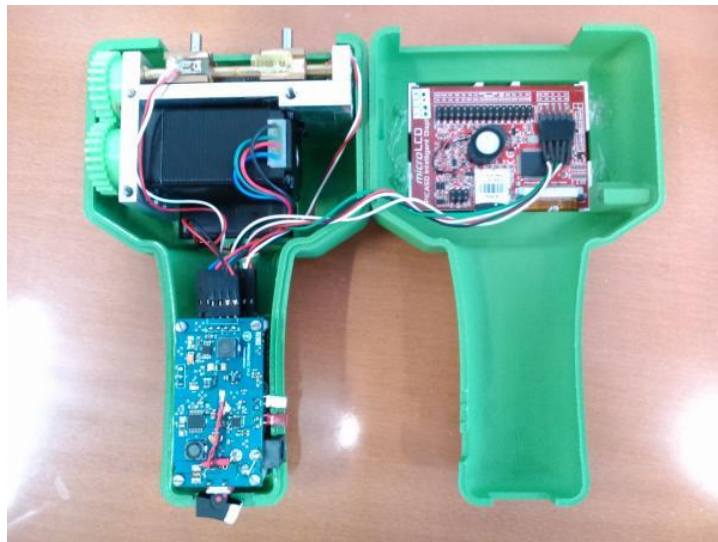


Figura 10: Electrónica de control Pressmatic.

La funcionalidad y versatilidad de Pressmatic se basa en la capacidad de adaptarse a la situación que se presente frente al individuo y en la resolución de tareas cotidianas de manera eficaz.

En la figura 11 (inferior) se muestran las herramientas de la máquina (izquierda), que gracias a la base común encajan en el tronco y proveen al dispositivo la disposición a diferentes situaciones; a la derecha, se puede ver la ejecución de corte del instrumento con una gran precisión.



Figura 11: Herramientas de Pressmatic

Para avanzar en el control del dispositivo, este trabajo se ha centrado en la programación y depuración de la segunda versión de la aplicación para Smartphone, mejorando su diseño, funcionalidad y capacidad de interacción con el usuario. Para lo cual, se ha añadido un sintetizador de voz bilingüe, capacidades de uso y una interfaz atractiva y dinámica adaptada a dispositivos con sistema operativo Android.

En la programación a bajo nivel se ha mantenido la forma de conexión por Bluetooth para adaptar la aplicación a los comandos utilizados en la máquina de estados que genera los movimientos automáticos.

Para el control por voz se ha utilizado una de las herramientas de las librerías Android para captar y transformar en texto los comandos de voz. Luego, comparar las palabras capturadas con unas elegidas para la ejecución de tareas.

Android ha sido el entorno elegido para continuar la construcción de la aplicación de control. Fue diseñado para ser un Sistema Operativo (como Windows o IOS) de núcleo Linux para dispositivos móviles inteligentes, aunque se ha convertido también en una plataforma software para desarrolladores. Tras ser adquirido por Google, se implementaron bibliotecas en lenguaje Java para posibilitar el control de dispositivos.

Las aplicaciones operan en un framework Java (Android Studio, Eclipse...) mediante programación orientada a objetos con bibliotecas escritas en lenguaje C que poseen:



- Un administrador para la interfaz gráfica de usuario “Surface manager”.
- Un framework “OpenCore” (libre).
- Una base de datos relacional.
- Una interfaz de programación gráfica en código base Android.
- Un motor de renderizado, un motor gráfico y una biblioteca estándar de *C Bionic*.

Además, es una plataforma escrita en código abierto, por lo que cualquier desarrollador puede crear aplicaciones y comercializarlas en su espacio de venta *Google Play Store*, siempre que las compile en el código nativo de Android ARM.

Su principal competidor, IOS (sistema operativo de Apple), sólo se ejecuta en los dispositivos de la compañía, mientras que Android, es libre y cualquier compañía tecnológica o usuario puede implementar código y adaptarlo a la mayoría de los dispositivos. Esta es una de las principales ventajas del sistema Android respecto a los demás.

Para mayor versatilidad, Android dispone de multitud de funcionalidades diseñadas por los desarrolladores de Google y de uso libre para los interesados, como el traductor, la cámara o el sintetizador de voz, que en este proyecto se ha utilizado para la aplicación y así poder realizar las tareas que posibilita Pressmatic con menor dependencia manual.

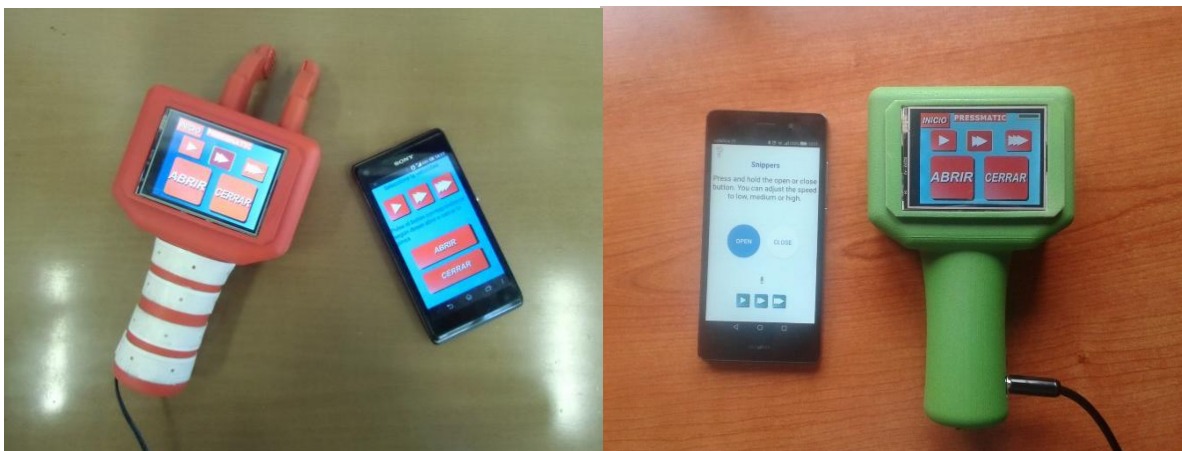


Figura 12: Comparación aplicaciones para Pressmatic



3.2.1 Descripción de las características funcionales

La aplicación ha sido construida de forma dinámica, de modo que, es posible moverse por los diferentes bloques de la aplicación en cualquier momento.

A continuación, se muestra el diagrama de casos de uso. Está formado por 4 bloques principales:

1. Instrucciones
2. Conectividad
3. Elección de herramienta (con su correspondiente funcionalidad)
4. Lenguaje

Para seleccionar una herramienta de trabajo es necesario estar conectado al dispositivo, para lo que se necesita activar el Bluetooth del teléfono. Si no está activo, aparece un mensaje de alerta comunicándolo.

Es posible cambiar el idioma de la aplicación o acceder a la ayuda desde cualquiera de los bloques antes mencionados, lo que facilita al usuario desenvolverse por la aplicación.

Si se desconecta el Bluetooth del teléfono una vez conectado a Pressmatic, la aplicación lo detectará y volverá a aparecer la pantalla de activación, tras lo que será posible conectarse de nuevo al aparato y seleccionar una de las herramientas.

En la siguiente figura (figura 13) se puede ver el diagrama de casos de uso, en el que se han agrupado las actividades de las instrucciones y herramientas en un mismo bloque para facilitar su lectura.

Al acceder a cualquiera de ellas siempre empieza por la primera actividad del grupo, no es posible que el usuario decida si quiere saltar directamente a la pantalla tres del tutorial de ayuda o a la segunda herramienta.

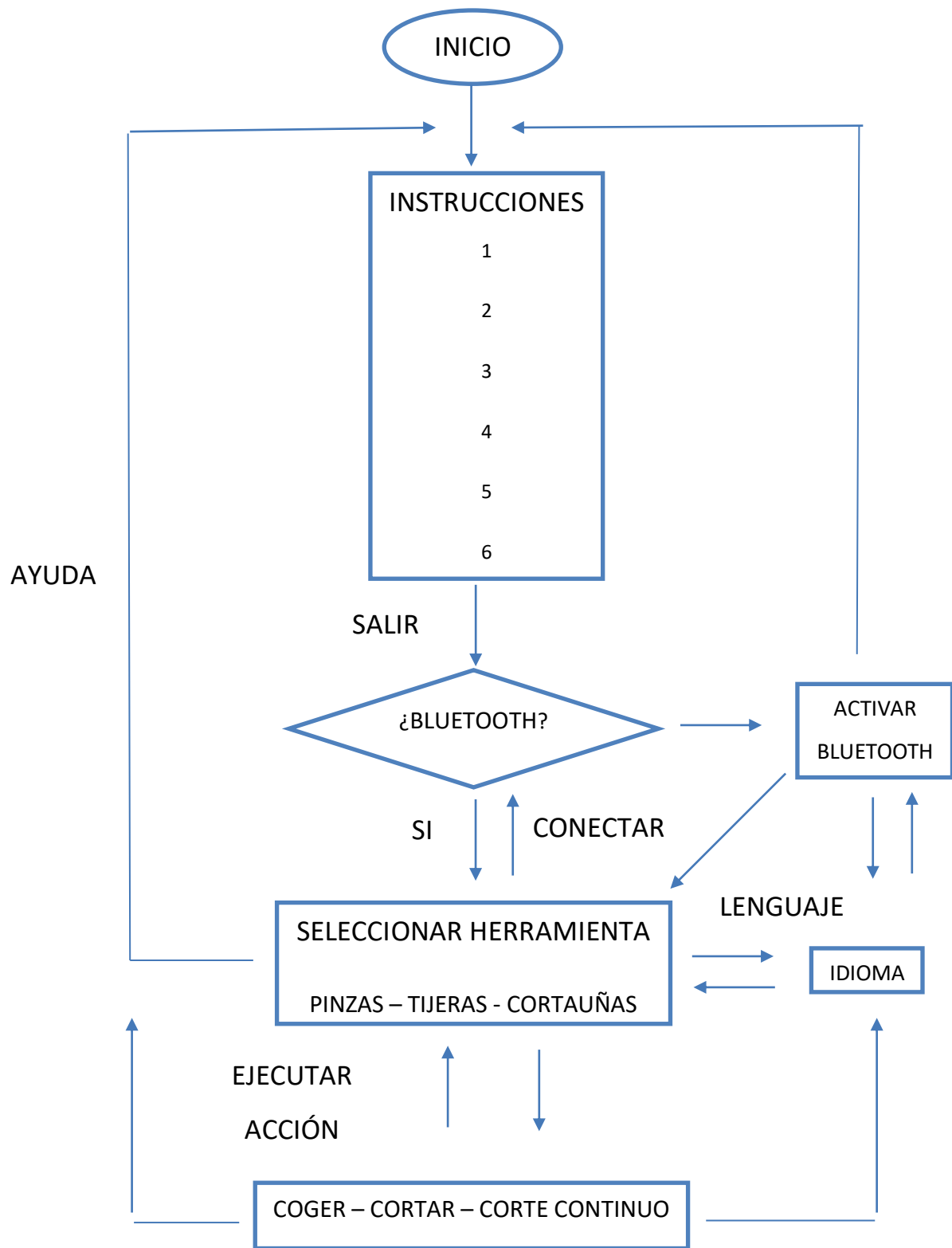


Figura 13: Diagrama de flujo



3.2.2 Restricciones del sistema

Es necesario analizar las limitaciones del sistema a implementar y de los accesorios disponibles para entender el alcance y los inconvenientes que pueden surgir en el desarrollo del trabajo.

Para cambiar las herramientas, se requiere la intervención de una persona para retirarlas y encajarlas (no es un proceso automático), lo que ralentiza las operaciones.

Las principales restricciones del sistema Android y de la aplicación desarrollada son:

- La configuración del dispositivo para habilitar las funcionalidades, como el reconocimiento de voz, que necesita de conexión a internet o habilitar la opción on-line en los ajustes del teléfono.
- Los inconvenientes a la hora de conectar los dispositivos (errores de conexión, tiempo de respuesta, diferencias en la funcionalidad según marca de terminal y versión...).
- El funcionamiento base de la aplicación es el envío de una orden a Pressmatic para llevar a cabo una acción o cambio en el dispositivo. Estas órdenes se envían a través del protocolo bluetooth, y el envío en un periodo de tiempo demasiado corto o demasiadas órdenes provoca problemas en la conexión y el funcionamiento de los botones.

EL software ha sido construido sobre una aplicación ya desarrollada, por lo que ha sido necesario mantener una versión mínima por debajo de la habitual para que el IDE fuese capaz de reconocer los paquetes.

Para llevar a cabo el control del dispositivo Pressmatic hay que establecer una conexión por Bluetooth, por lo que sólo es posible enlazar un aparato en cada momento y enviar las órdenes de una en una. Además, la tecnología Bluetooth tiene un límite de separación entre dispositivos de 10 metros (al superarla, se pierde la conexión) y en el tiempo de conexión establecida cuando el terminal está en reposo (no se envían órdenes).



3.2.3 Entorno operacional

El entorno operativo incluye el sistema operativo y la interfaz de usuario asociada al sistema y a las aplicaciones nativas. Para la realización de este proyecto han sido necesarios elementos hardware (físicos) y software (para desarrollar la aplicación).

A continuación, se detalla el uso de dichos elementos, teniendo en cuenta el papel que han desempeñado en la ejecución del proyecto. Aunque, gracias al avance de las tecnologías, puede que sea posible llevar a cabo el desarrollo y la implementación de un nuevo proyecto sobre el llevado a cabo en un mismo dispositivo, como puede ser una Tablet o un Note Book:

- Un ordenador con sistema operativo Ubuntu 12.04 o superior: con al menos 4Ghz de memoria RAM y procesador Intel i5 o ARM para poder ejecutar el emulador de Android y el software de implementación con la mayor celeridad posible y sin inconvenientes debido al hardware utilizado.
- El IDE de desarrollo Android Studio, con las herramientas de Android SDK necesarias y el entorno Java con la versión adecuada para la instalación de los paquetes: este entorno de desarrollo necesitará al menos 3 horas para completar su instalación (desde la descarga del software de la página oficial hasta la instalación de las herramientas y bibliotecas necesarias).
- Un teléfono móvil con sistema operativo Android: preferiblemente dos o más terminales, con diferentes versiones y capacidad del procesador, para poder asegurar el funcionamiento de la aplicación para una amplia gama de dispositivos.
- El dispositivo asistencial Pressmatic: provisto e implementado por el departamento de Robótica de la Universidad Carlos III de Madrid, permite llevar a cabo acciones complejas a personas con minusvalía manual.



3.2.4 Especificación de casos de uso

Para la descripción de cualquier trabajo o proyecto desarrollado es necesario realizar un estudio de las especificaciones de las acciones que ha de tomar el sistema y el orden en que deben sucederse. Para ello, en este apartado se detallan los pasos y las especificaciones en cada uno de ellos que tiene que seguir la aplicación (dados en las condiciones necesarias para el desarrollo del proyecto).

La aplicación comienza mostrando la interfaz de inicio, tras lo que aparecen las instrucciones. El usuario puede desplazarse a izquierda o derecha y, cuando desee conectarse con el dispositivo, debe pulsar el botón “X” para abandonar las instrucciones:

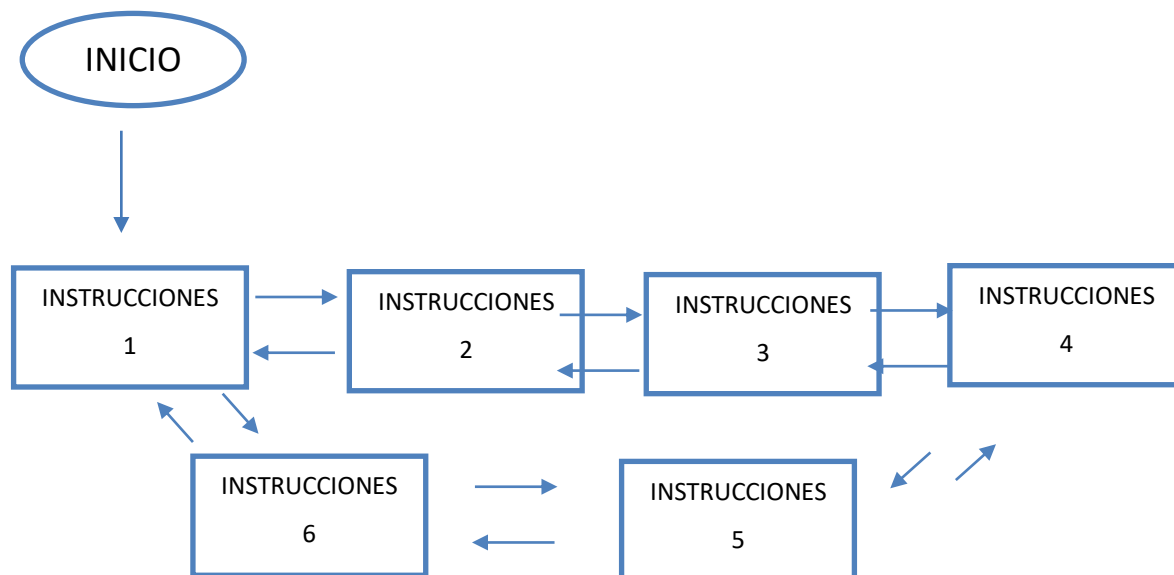


Figura 14: Especificación instrucciones.

Seguidamente, si no está habilitado el Bluetooth del terminal, aparecerá una pantalla con un mensaje de aviso de la necesidad de hacerlo, con dos botones para elegir activarlo o no. Si se elige “no”, al intentar conectar con Pressmatic volverá a indicar la necesidad de habilitar el Bluetooth y entrará en un bucle hasta que el usuario decida o activarlo manualmente o indicar “sí” cuando aparezca la pantalla de activación nuevamente.

Desde estas actividades, es posible cambiar el idioma de la aplicación o volver al menú de ayuda. En la siguiente figura (figura 15) se puede ver el diagrama de uso de la activación del bluetooth de forma detallada:

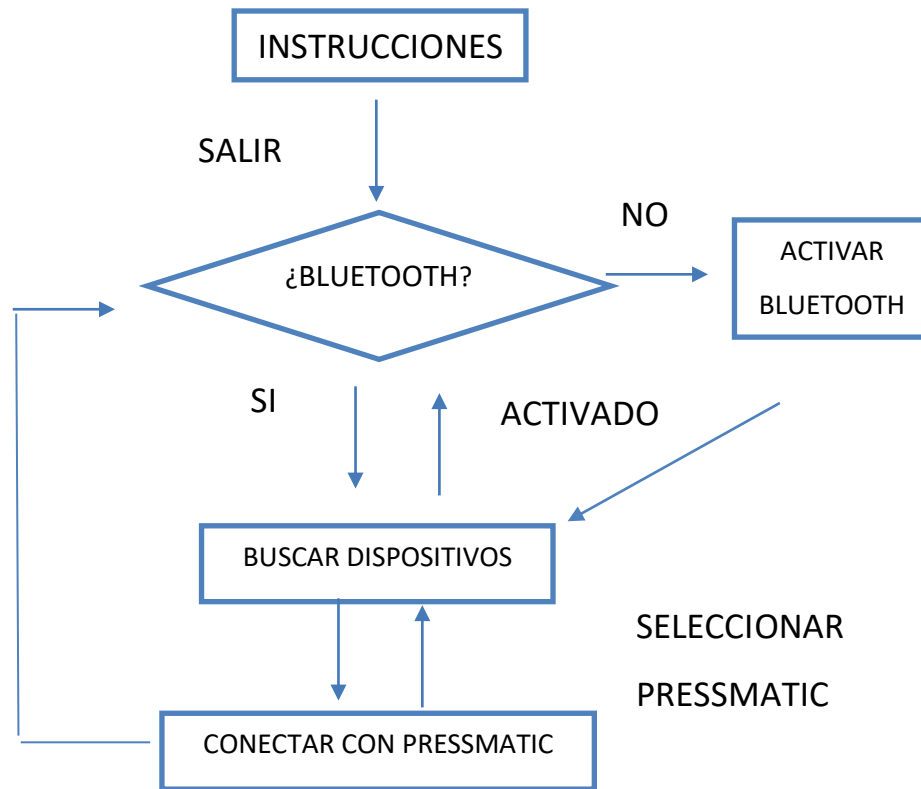


Figura 15: Especificaciones conexión bluetooth.

Tras conectar la aplicación con el dispositivo Pressmatic, se ha de elegir una herramienta acorde con el utensilio colocado físicamente en el aparato. El usuario puede moverse de izquierda a derecha buscando el elemento que necesita, además de tener la capacidad de cambiar el idioma y de volver a consultar las instrucciones.

Tras seleccionar un utensilio, aparecerá una nueva pantalla con la acción a realizar con cada uno de ellos. En todas ellas, existe la posibilidad de modificar la velocidad de acción en cada orden enviada desde el terminal al dispositivo.

En la siguiente imagen (figura 16) se puede apreciar el funcionamiento del sistema para llevar a cabo los trabajos necesarios (se puede ver la comprobación continua y necesaria de la conexión entre el terminal y el dispositivo):

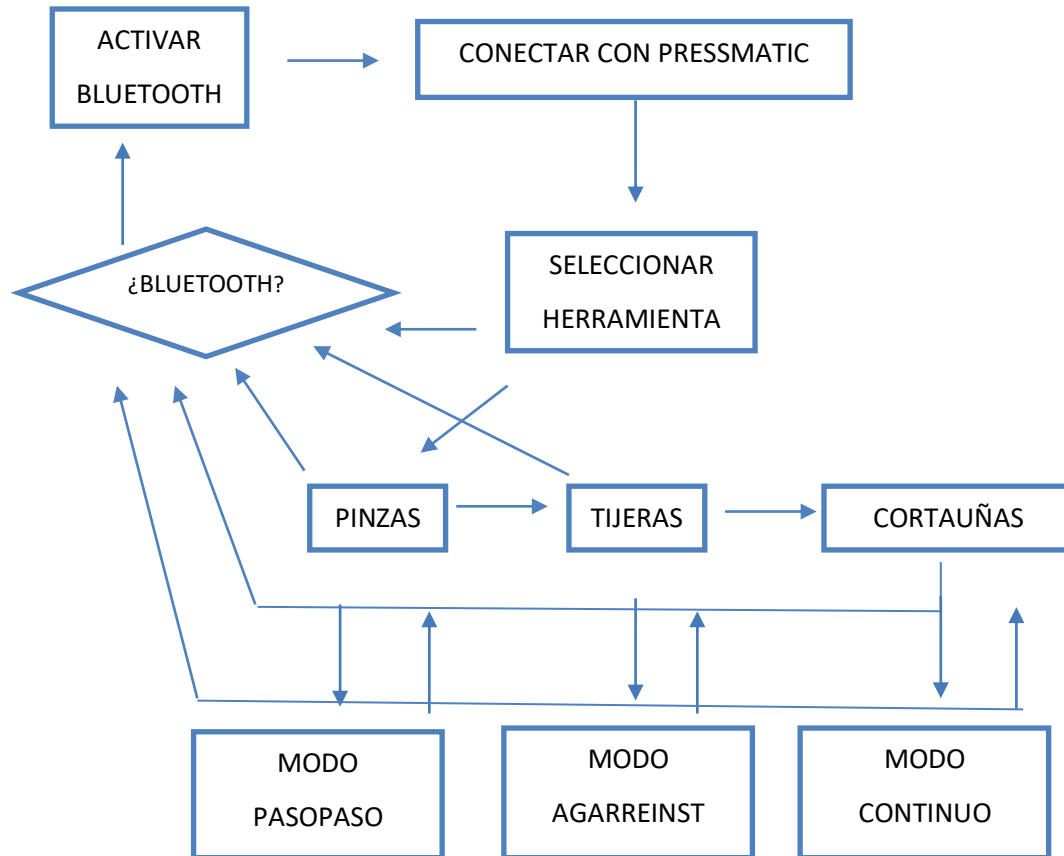


Figura 16: Especificaciones uso de herramientas.

El gobierno del dispositivo se sustenta en el envío de comandos, que se envían tras el pulsado de ciertos botones de la app, correspondientes a acciones del dispositivo (selección herramientas, abrir, cerrar, cortar, etcétera). El comando correspondiente a cada una de las acciones que se encuentra en el dispositivo se incluye en el socket utilizado para enviar mensajes.

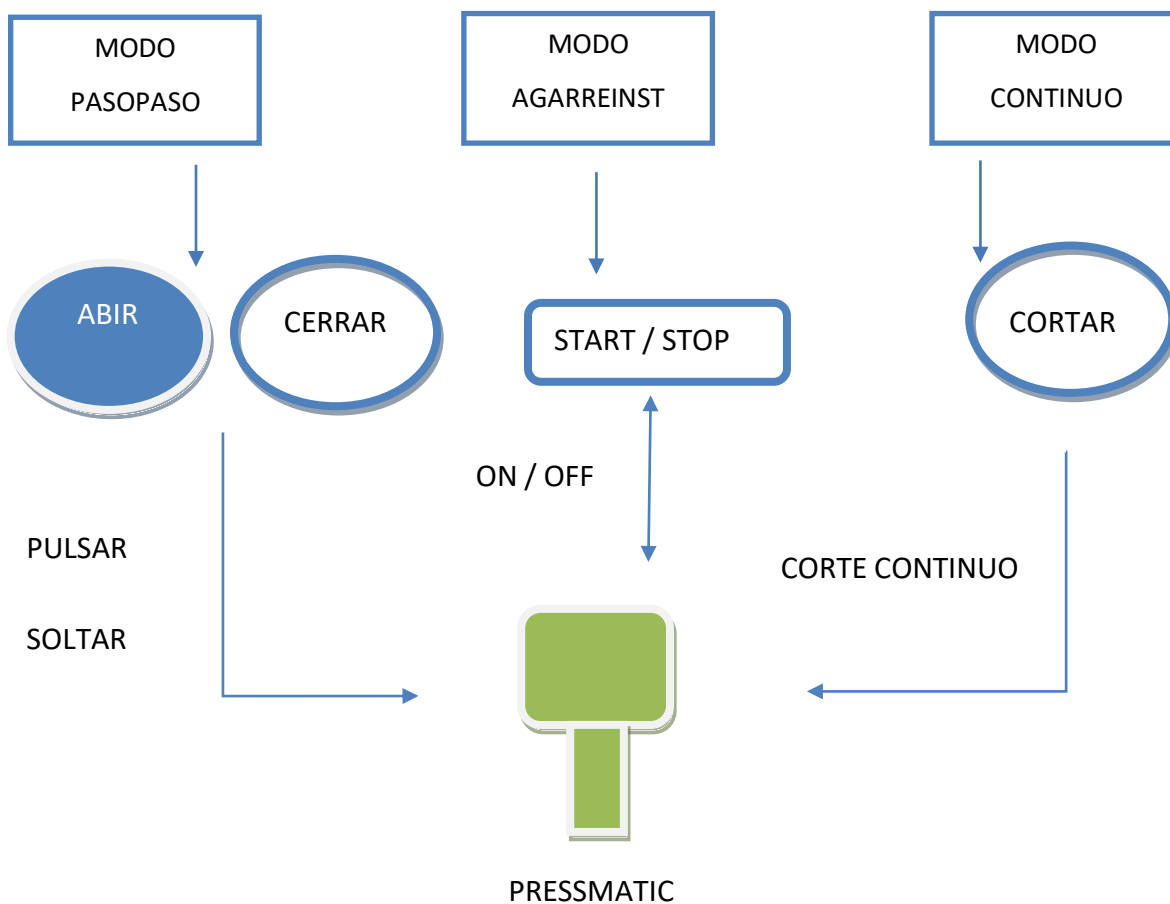


Figura 16: Especificaciones órdenes a Pressmatic.

Para llevar a cabo esta nueva implementación de la aplicación y el cambio en la interfaz de usuario, ha sido necesaria la modificación del código original y el aumento de métodos y clases para poder ajustar la interfaz al diseño requerido.

Además de una mejora del código, es necesario conocer en profundidad las restricciones y funcionalidades de los elementos que conforman el trabajo. En el siguiente apartado, se da información acerca de los aparatos utilizados de forma más detallada que anteriormente.



3.2.4.1 Descripción de los actores

Para llevar a cabo este proyecto han sido necesarios elementos hardware y software que, al combinarlos, es posible reproducir un sistema de operación controlando la ejecución de una herramienta con un terminal Android.

Estos elementos han sido seleccionados de forma que permitan el desarrollo del proyecto eficazmente con el presupuesto necesario para cada elemento.

- Dispositivo electromecánico Pressmatic: proyecto perteneciente al *Robotics Lab* de la Universidad Carlos III de Madrid y cuya mejora de funcionamiento y flexibilidad es tarea en este trabajo.
 - Compuesto de una batería, una pantalla táctil que funciona de interfaz de usuario y la electrónica de control capaz de generar el movimiento de los utensilios.
 - Combinado con las tres herramientas elegidas en su construcción: unas tijeras y unas pizas grandes y pequeñas.
- Teléfono móvil inteligente con sistema operativo Android: se han utilizado 2 terminales para el desarrollo del proyecto (sin contar con el emulador del IDE de desarrollo).
 - Huawei P8 Little: con versión del sistema operativo Android 6.0, 2.00Gb de memoria RAM, procesador de 1.2Ghz y versión “Kernel” 3.10.86.
 - Samsung Galaxy Trend: versión de Android 4.0.4, versión “Kernel” 3.0.8
- Ordenador con sistema operativo Ubuntu Linux 12.04 o superior: Toshiba Satellite S50, procesador Intel Core i7, procesador gráfico AMD “Radeon” y 8Gb de memoria RAM.
- IDE de desarrollo oficial Android Studio (última versión disponible).



3.2.4.2 Descripción de los atributos de los casos de uso

En este apartado se identifica cada uno de los casos de uso del sistema con un atributo identificativo para clarificar su descripción textual. En los siguientes puntos está detallado el significado de cada uno de los atributos utilizados (teniendo en cuenta las descripciones usualmente utilizadas para detallar los casos de funcionamiento de sistemas similares):

- **Código:** Identificación abreviada del caso de uso, se forma por las siglas CU (caso uso) seguidas de un guion y de tres dígitos (CU-001).
- **Nombre:** Identificación extendida del caso de uso que proporciona información acerca de la tarea que se realiza en el apartado.
- **Actores:** Conjunto de elementos que actúan en el caso de uso. El caso de uso representa una funcionalidad requerida por un actor.
- **Descripción:** Explicación detallada de la funcionalidad del caso de uso.
- **Condiciones:** Inventario de los requisitos que deben cumplirse para poder realizar una operación, y el estado en el que queda el sistema tras realizar una operación. Se dividen en condiciones anteriores a que se dé la tarea y posteriores a su realización.
- **Escenario:** Especificación de las acciones que se ejecutaran en el caso de uso, paso a paso de forma descriptiva.

| Caso de uso | |
|-----------------------|--------|
| Código | CU-000 |
| Nombre | |
| Actores | |
| Descripción | |
| Precondiciones | |
| Poscondiciones | |
| Escenario | |

Tabla 1 – Tabla de casos de uso



3.2.4.3 Descripción textual de los casos de uso

| Caso de uso | |
|-----------------------|--|
| Código | CU-001 |
| Nombre | Inicio de la aplicación |
| Actores | Dispositivo Pressmatic y terminal con sistema operativo Android |
| Descripción | Se inicia la aplicación Pressmatic pulsando en el icono del terminal |
| Precondiciones | Tener el dispositivo Pressmatic encendido |
| Poscondiciones | Comienzo de la aplicación, muestra de la interfaz principal. |
| Escenario | <p>Se enciende el dispositivo Pressmatic, con batería cargada o conectado a red.</p> <p>Están preparados los utensilios a utilizar.</p> <p>En el terminal, localizar la aplicación de Pressmatic (previamente instalada).</p> <p>La aplicación Pressmatic muestra la pantalla de inicio en el dispositivo móvil.</p> |

Tabla 2 – CU-001

| Caso de uso | |
|-----------------------|---|
| Código | CU-002 |
| Nombre | Ayuda |
| Actores | Terminal con aplicación Pressmatic |
| Descripción | Tras la pantalla de inicio aparecen las instrucciones de la aplicación |
| Precondiciones | Acceder a la aplicación desde el terminal |
| Poscondiciones | Guía para el usuario, instrucciones de uso |
| Escenario | Tras ejecutar la aplicación el usuario tiene una ayuda de cómo utilizarla |

Tabla 3 – CU-002



| Caso de uso | |
|-----------------------|--|
| Código | CU-003 |
| Nombre | Conectividad |
| Actores | Terminal |
| Descripción | La aplicación ejecuta mensajes de aviso para activar el Bluetooth |
| Precondiciones | Salir de las instrucciones |
| Poscondiciones | Conectar con el dispositivo Pressmatic |
| Escenario | <p>Tras entender el funcionamiento de la aplicación, es necesario activar el Bluetooth.</p> <p>Si ya está activado, no aparecerá ningún mensaje de alerta.</p> |

Tabla 4 – CU-003

| Caso de uso | |
|-----------------------|---|
| Código | CU-004 |
| Nombre | Conectar con Pressmatic |
| Actores | Dispositivos Pressmatic y móvil |
| Descripción | Se conecta el teléfono a Pressmatic mediante Bluetooth |
| Precondiciones | Bluetooth del teléfono activado y Pressmatic encendido |
| Poscondiciones | Conexión realizada con éxito |
| Escenario | <p>Tras habilitar el Bluetooth del teléfono, se pulsa el botón conectar de la aplicación.</p> <p>El dispositivo entonces buscará las redes Bluetooth disponibles, el usuario debe elegir la de Pressmatic.</p> <p>Tras conectarse, aparecerá un mensaje de aviso y se podrá elegir una herramienta.</p> |

Tabla 5 – CU-004



| Caso de uso | |
|-----------------------|---|
| Código | CU-005 |
| Nombre | Idioma |
| Actores | Terminal Android |
| Descripción | Es posible cambiar el idioma de la aplicación de español a inglés |
| Precondiciones | Salir de las instrucciones de ayuda |
| Poscondiciones | Idioma elegido seleccionado para la aplicación |
| Escenario | <p>Si la aplicación se encuentra en un idioma distinto al del usuario, es posible cambiarlo pulsando el icono de la rueda (arriba a la izquierda).</p> <p>Para ello, hay que salir de la ayuda, después de cambiar el idioma al volver a las instrucciones aparecerán en el lenguaje elegido.</p> |

Tabla 6 – CU-005

| Caso de uso | |
|-----------------------|---|
| Código | CU-006 |
| Nombre | Elección de herramienta |
| Actores | Dispositivos Pressmatic y Android |
| Descripción | Se elige una herramienta para realizar la tarea deseada |
| Precondiciones | Conectar el dispositivo Android con Pressmatic mediante Bluetooth |
| Poscondiciones | Tras seleccionar un elemento, aparecerá una pantalla con la acción posible |
| Escenario | <p>Al conectar el terminal Android con Pressmatic aparece una nueva pantalla para elegir una herramienta, ésta debe coincidir con la que se coloque en el dispositivo.</p> <p>Es necesario coger uno de los utensilios disponibles y encajarlo manualmente en el aparato.</p> |

Tabla 7 – CU-006



| Caso de uso | |
|-----------------------|--|
| Código | CU-007 |
| Nombre | Ejecución de tarea |
| Actores | Dispositivos Pressmatic y Android |
| Descripción | Se lleva a cabo la acción deseada según la herramienta elegida |
| Precondiciones | Colocar una herramienta en el dispositivo y elegirla en la aplicación |
| Poscondiciones | Realización de la tarea |
| Escenario | <p>Después de seleccionar una herramienta en la aplicación y encajar manualmente la misma en el dispositivo, es posible realizar las tareas.</p> <p>Se agarra Pressmatic con una mano y con la otra se activa el botón de acción, es posible modificar la velocidad de trabajo en la aplicación.</p> |

Tabla 8 – CU-007

| Caso de uso | |
|-----------------------|--|
| Código | CU-008 |
| Nombre | Cambio de utensilio |
| Actores | Dispositivos Pressmatic y Android |
| Descripción | Pueden sustituirse los elementos de trabajo de forma indefinida |
| Precondiciones | Dispositivo Pressmatic y Android conectados a través de Bluetooth |
| Poscondiciones | Herramienta elegida en la aplicación y colocada en Pressmatic manualmente |
| Escenario | <p>Tras trabajar con una herramienta, es posible retirarla del dispositivo Pressmatic y colocar otra distinta.</p> <p>Se elige una nueva herramienta en la aplicación y se ejecuta la tarea que con el nuevo elemento sea posible.</p> |

Tabla 9 – CU-008



3.2.4.2 Descripción de los atributos de los requisitos

En este apartado se identifica cada uno de los requisitos del sistema (sus funcionalidades o restricciones) con un atributo para clarificar su descripción textual.

- Código: Identificación abreviada del requisito, que será dividido en funcional (código RF) y no funcional (RFN). Por ejemplo, RF-001.
- Nombre: Identificación extendida del requisito.
- Descripción: Explicación detallada del requisito.
- Fuente: Indica de dónde ha sido identificado el requisito.
- Necesidad: Nivel de implementación del requisito. Puede tomar los tres valores:
 - Esencial: Debe que ser implementado.
 - Deseable: Es preferible, pero no obligatorio.
 - Opcional: Puede o no desarrollarse.
- Prioridad: Determina la importancia del requisito, su orden en el diseño y en la implementación. Puede tomar los tres valores:
 - Alta: Debe ser implementado en las fases iniciales del progreso.
 - Media: Debe ser implementado tras los requisitos de alta prioridad.
 - Baja: Debe ser implementado en las fases finales del desarrollo (no influye en el funcionamiento del sistema).
- Estabilidad: Define la consistencia del requisito y si podrá ser modificado. Los valores que puede tomar son:
 - Estable: No puede variar.
 - Inestable: Varía.
- Verificabilidad: Muestra de que forma el requisito es incluido en el sistema. Los valores que puede tomar son:
 - Alta: Se verifica fácilmente si el requisito está incorporado en el sistema.
 - Media: Es necesaria una comprobación compleja para identificar el requisito.
 - Baja: Es difícil verificarlo o no es posible.



3.2.5 Especificación de requisitos

Los requisitos se van a organizar en una tabla como la que se muestra a continuación, agrupando código y fuente, necesidad y prioridad y estabilidad y verificabilidad en la misma línea (para facilitar la comprensión).

| Requisito del sistema | | | |
|-----------------------|--|-----------------|--|
| Código | | Fuente | |
| Nombre | | | |
| Descripción | | | |
| Necesidad | | Prioridad | |
| Estabilidad | | Verificabilidad | |

Tabla 10 – Tabla de requisitos.

Los requisitos se clasifican como funcionales o como no funcionales. Los requisitos funcionales son imprescindibles para el correcto “funcionamiento” de la aplicación, pues sin la ejecución de uno de ellos no será posible el desarrollo de la tarea y aparecerá un mensaje de error; los requisitos no funcionales, pueden o no llevarse a cabo, pues no impiden el desarrollo de las tareas ni el correcto funcionamiento de la aplicación.

3.2.5.1 Requisitos funcionales

| Requisito del sistema | | | |
|-----------------------|--|-----------------|---------|
| Código | RF-001 | Fuente | Usuario |
| Nombre | Activación de Pressmatic | | |
| Descripción | El usuario encenderá el dispositivo y preparará las herramientas | | |
| Necesidad | Esencial | Prioridad | Alta |
| | | | |
| Estabilidad | Estable | Verificabilidad | Alta |

Tabla 11 – RF-001



| Requisito del sistema | | | |
|-----------------------|---|------------------------|---------|
| Código | RF-002 | Fuente | Usuario |
| Nombre | Iniciar la aplicación | | |
| Descripción | Habrá un icono en el terminal Android del usuario para comenzar la aplicación | | |
| Necesidad | Esencial | Prioridad | Alta |
| Estabilidad | Estable | Verificabilidad | Alta |

Tabla 12 – RF-002

| Requisito del sistema | | | |
|-----------------------|---|------------------------|------------------|
| Código | RF-003 | Fuente | Terminal Android |
| Nombre | Requisitos de la aplicación | | |
| Descripción | La aplicación mostrará las pantallas de ayuda y avisará si no está conectado el Bluetooth del dispositivo | | |
| Necesidad | Esencial | Prioridad | Alta |
| Estabilidad | Estable | Verificabilidad | Alta |

Tabla 13 – RF-003

| Requisito del sistema | | | |
|-----------------------|---|------------------------|---------|
| Código | RF-004 | Fuente | Usuario |
| Nombre | Conexión con Pressmatic | | |
| Descripción | El usuario debe conectar la aplicación con el dispositivo por Bluetooth | | |
| Necesidad | Esencial | Prioridad | Alta |
| Estabilidad | Inestable | Verificabilidad | Media |

Tabla 14 – RF-004



| Requisito del sistema | | | |
|-----------------------|--|------------------------|---------|
| Código | RF-004 | Fuente | Usuario |
| Nombre | Mantener la conexión | | |
| Descripción | El usuario debe vigilar la conexión entre el dispositivo y el terminal | | |
| Necesidad | Opcional | Prioridad | Baja |
| Estabilidad | Inestable | Verificabilidad | Media |

Tabla 15 – RF-004

| Requisito del sistema | | | |
|-----------------------|--|------------------------|---------|
| Código | RF-005 | Fuente | Usuario |
| Nombre | Elección de herramienta | | |
| Descripción | El usuario deberá elegir el mismo utensilio físicamente y en la aplicación | | |
| Necesidad | Esencial | Prioridad | Alta |
| Estabilidad | Estable | Verificabilidad | Alta |

Tabla 16 – RF-005

| Requisito del sistema | | | |
|-----------------------|---|------------------------|------------------------|
| Código | RF-006 | Fuente | Dispositivo Pressmatic |
| Nombre | Ejecución de tarea | | |
| Descripción | El dispositivo llevará a cabo la acción indicada por el usuario | | |
| Necesidad | Esencial | Prioridad | Alta |
| Estabilidad | Estable | Verificabilidad | Alta |

Tabla 17 – RF-006



| Requisito del sistema | | | |
|-----------------------|--|------------------------|---------|
| Código | RF-007 | Fuente | Usuario |
| Nombre | Mantener la conexión bluetooth | | |
| Descripción | El usuario debe vigilar la conexión bluetooth del terminal Android | | |
| Necesidad | Opcional | Prioridad | Baja |
| Estabilidad | Inestable | Verificabilidad | Media |

Tabla 18 – RF-007

3.2.5.2 Requisitos no funcionales

| Requisito del sistema | | | |
|-----------------------|---|------------------------|---------|
| Código | RNF-001 | Fuente | Usuario |
| Nombre | Idioma | | |
| Descripción | El usuario podrá cambiar el lenguaje de la aplicación | | |
| Necesidad | Opcional | Prioridad | Baja |
| Estabilidad | Inestable | Verificabilidad | Alta |

Tabla 19 – RNF-001

| Requisito del sistema | | | |
|-----------------------|---|------------------------|---------|
| Código | RNF-002 | Fuente | Usuario |
| Nombre | Reconocimiento de voz | | |
| Descripción | El usuario podrá navegar por la aplicación utilizando el sintetizador | | |
| Necesidad | Opcional | Prioridad | Baja |
| Estabilidad | Inestable | Verificabilidad | Alta |

Tabla 20 – RNF-002



| Requisito del sistema | | | |
|-----------------------|---|------------------------|----------|
| Código | RNF-003 | Fuente | Terminal |
| Nombre | Sistema operativo | | |
| Descripción | El terminal Android podrá utilizar una versión 3.0 o superior | | |
| Necesidad | Opcional | Prioridad | Baja |
| Estabilidad | Estable | Verificabilidad | Alta |

Tabla 21 – RNF-003

Una vez conocidos los requisitos del Sistema, se puede observar que no es complicado diferenciar qué funcionalidades de la aplicación son necesarias para su correcto desarrollo y cuáles son implementaciones añadidas para mejorar la calidad y el atractivo de la aplicación.

La importancia de este apartado se debe a que en todo proyecto es necesario detallar los puntos que ha de seguir en cada fase para llegar al objetivo deseado. En la ingeniería de sistemas, un requisito es imprescindible para documentar los contenidos y necesidades o funcionalidades del sistema. Incluyen tanto especificaciones de acción como respecto al entorno operacional.

Usualmente en ingeniería, los requisitos se referían a los datos iniciales de entrada en la etapa de diseño. Hacían referencia a lo que debería hacer el sistema, pero no a los pasos que debería llevar a cabo para cumplir su objetivo.

Ahora, se detalla el diseño de la plataforma Android utilizada en este proyecto, desde las capas interiores del sistema operativo hasta la visualización de aplicaciones del usuario. Además, se describe la arquitectura que posee el sistema y se explica punto por punto su nivel y utilidad en el desarrollo del lenguaje.



3.3 Diseño del sistema

Android posee una *Interfaz de Programación de Aplicaciones* (API) que incorpora medios que posibilitan la transmisión de información entre diferentes terminales o a través de la red.

Aunque, dependiendo del hardware del dispositivo puede contener más o menos funcionalidades, se conservan tres tipos de comunicación estándar: red Wifi, telefonía y bluetooth.

1. Red: El paquete que proporciona el acceso a la pila de red Wifi es el “android.net.wifi”. Es capaz de notificar desde los puntos de acceso que ha detectado hasta el estado de la conexión. También incluye métodos para escanear, iniciar, finalizar y configurar conexiones, etcétera.
2. Telefonía: La API “android.telephony” permite acceder a la información base del teléfono (tipo de red, estado de la conexión...). Las principales clases que participan en este proceso son:
 - a. “NeighboringCellInfo”: obtiene información de la celda próxima.
 - b. “PhoneNumberUtils”: permite administrar los “String”.
 - c. “PhoneStateListener”: dirige los cambios de estado del teléfono.
 - d. “SMSManager” y “SMSMessage”: operaciones y mensajes SMS.
 - e. “TelephonyManager”: Permite el acceso a los servicios de telefonía del terminal.
3. Bluetooth: La pila de red Bluetooth es compatible con el sistema Android, por lo que es posible que un dispositivo pueda enviar y recibir datos con otros que posean esta tecnología.

A través del framework de la aplicación el usuario puede utilizar estas funciones de comunicación, mediante las denominadas Android Bluetooth API, que permiten a las aplicaciones conectarse de manera inalámbrica con otros dispositivos Bluetooth y habilitan las funciones inalámbricas punto a punto y de multipunto.

Las Bluetooth API dan la capacidad de implementar en una aplicación, entre otras, las siguientes opciones [19]:

- Consultar el adaptador local de Bluetooth y buscar dispositivos con esta tecnología
- Conectarse con otros dispositivos mediante el descubrimiento de servicios.
- Transferir datos entre dispositivos sincronizados.
- Administrar varias conexiones.

Para el diseño de la aplicación de este proyecto se ha mantenido la arquitectura base de la aplicación desarrollada anteriormente, cuya forma de comunicación con el dispositivo Pressmatic es mediante bluetooth. Es decir, utilizando las actividades, los bloques de conexión ya implementados y las órdenes de acción para enviar al dispositivo se ha reestructurado el software para adaptarlo al nuevo diseño requerido.

En la siguiente imagen (figura 17) se aprecia el funcionamiento de este protocolo de comunicación:

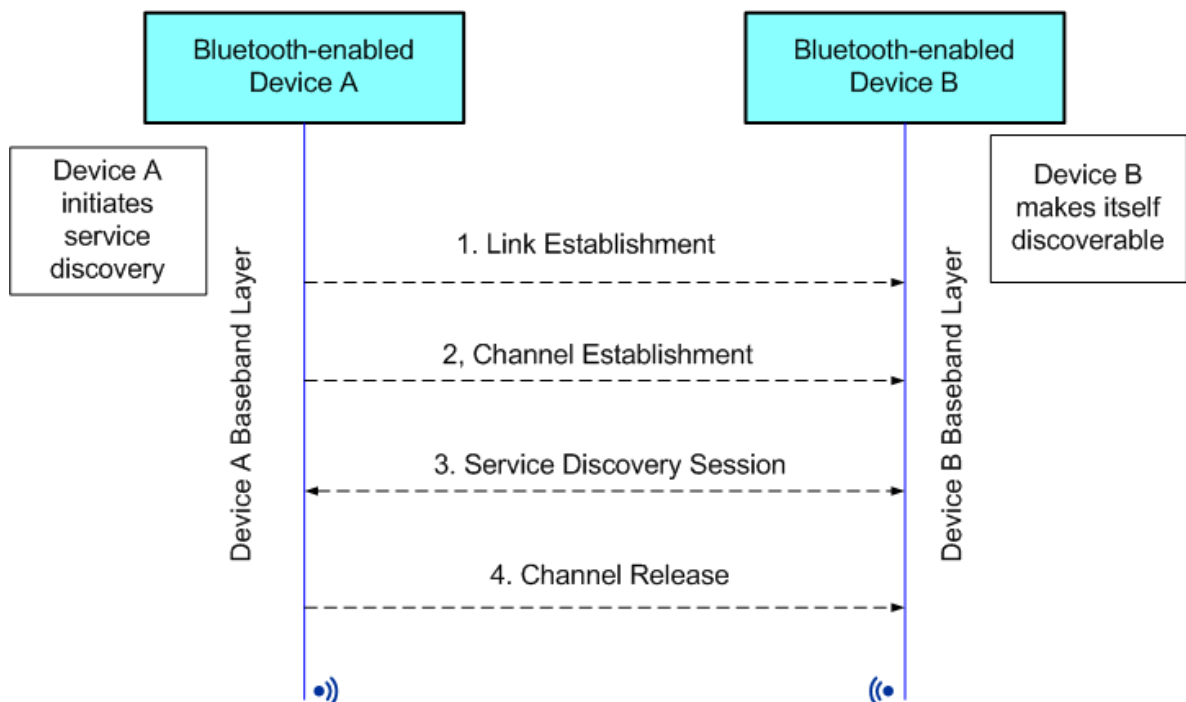


Figura 17: Comunicación bluetooth.



3.3.1 Arquitectura del sistema

Android es una plataforma ideada para que cada aplicación corra en un proceso independiente, instanciando una máquina virtual y un identificador de usuario (UID) individuales. Estos procesos transmiten información de forma concurrente, que lo convierte un sistema interoperable [15].

A continuación, en la figura 16, un diagrama de los principales componentes de la arquitectura de la plataforma muestra que Android se divide en cuatro capas. Desde la capa más baja, un núcleo “Kernel” Linux, hacia las capas superiores (las aplicaciones del sistema), interactuando entre ellas.

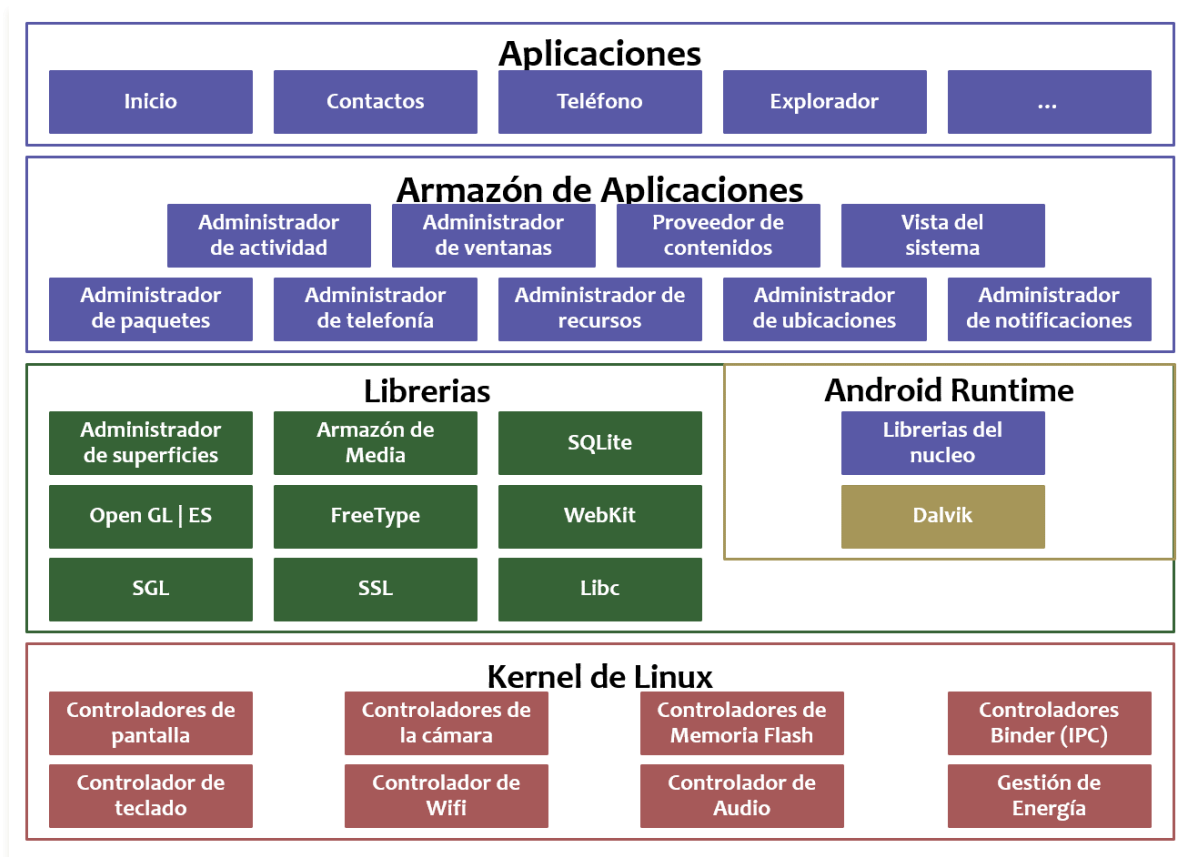


Figura 18: Arquitectura de la plataforma.

Ahora, se va a detallar cada uno de los campos de la arquitectura (mostrados en la figura anterior), desde la capa superior hasta el nivel inferior:



Figura 19: Arquitectura de la plataforma: Aplicaciones.

En el primer nivel aparecen las aplicaciones del sistema, ocupan la interfaz de usuario y pueden adquirirse a través de la red o estar por defecto en el terminal.



Figura 20: Arquitectura de la plataforma: Armazón de Aplicaciones.

En la siguiente capa (a un nivel más bajo), se encuentran utilidades que posee el dispositivo. A continuación, se explica cada una de las que aparece en la figura 20:

1. Administrador de actividad: administra el ciclo de vida de las aplicaciones del teléfono y provee un “Back stack” (un almacenamiento tras la interfaz vista de la aplicación) que permite al usuario ejecutar varias aplicaciones al mismo tiempo.
2. Administrador de ventanas: organiza las ventanas.
3. Proveedor de contenidos: tramita los datos comunes entre diferentes aplicaciones.
4. Vista del sistema: gestiona la interfaz de cada aplicación.
5. Administrador de notificaciones: permite al usuario elegir la forma de controlar eventos.
6. Administrador de ubicaciones: se encarga del posicionamiento del terminal.
7. Administrador de recursos: no está compuesto de código de programación, por lo que depende de cada aplicación.
8. Administrador de telefonía: tramita llamadas, mensajes...del teléfono en general.
9. Administrador de paquetes: se encarga de los paquetes de instalación/desinstalación para las aplicaciones.



Figura 21: Arquitectura de la plataforma: Librerías.

En un nivel superior al “Kernel” de Linux se encuentran las librerías utilizables para desarrolladores. Sirven para los gráficos, reproducción multimedia...y están implementadas en C/C++. El conjunto está formado por:

1. SGL (*Skia Graphics Library*): permite desarrollar figuras en dos dimensiones, está implementada en C++ y puede utilizarse en diferentes entornos (no solo dispositivos móviles).
2. SSL (*Secure Socket Layer*): suministra seguridad al dispositivo mediante el cifrado de datos (entre el servidor y el cliente). Android utiliza SSLv3.0 o SSLv2.0.
3. Libc (*Bionic*): se encarga de la gestión de excepciones de C++, de propiedades del sistema, sistema de logs...
4. OpenGL|ES (*OpenGL for Embedded Systems*): está basada en la API gráfica OpenGL y adaptada a dispositivos integrados, como teléfonos móviles y PDAs.
5. FreeType: administran mapas de bits y vectores.
6. WebKit: se utiliza para trabajar con aplicaciones para Safari o Google Chrome.
7. Surface Manager: Es la librería encargada de la gestión de superficies que gestiona el acceso al subsistema de la pantalla y compone capas de gráficos 2D y 3D desde múltiples aplicaciones.
8. Media Framework: librerías multimedia basadas en “Open CORE Packet Video”, admiten reproducción, grabación, ficheros de imagen estáticos, etcétera.
9. SQLite: permite almacenar datos en las aplicaciones. Funciona como motor para datos relacionales (transmitidos entre aplicaciones).



Figura 22: Arquitectura de la plataforma: Android Runtime.

En la misma capa se ejecuta el *Runtime* de Android, sobre la máquina virtual de Dalvik en las primeras versiones y después en cada aplicación al comienzo de su ejecución. Se encargan de optimizar el uso de memoria, batería, los procesos...

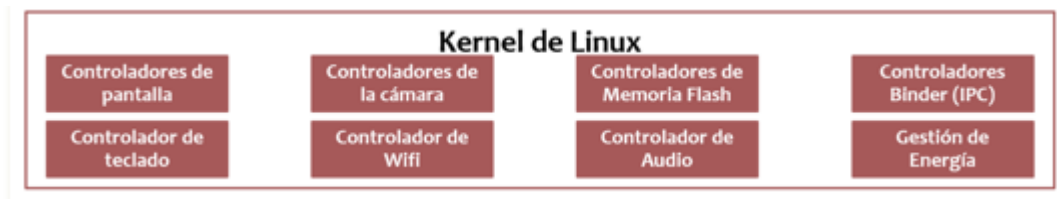


Figura 23: Arquitectura de la plataforma: Kernel de Linux.

Controla los drivers del sistema, los periféricos, servicios (seguridad, gestión de memoria, procesos...). Se actualiza con cada versión, según el “Kernel” Linux 2.6.x (siendo x variable) hasta Android 4.0, a partir de esta versión se utilizó el 3.0.x.

Linux es el núcleo o “Kernel” del Sistema Operativo libre GNU/Linux. Posee una licencia pública de GNU denominada “General Public License” (GPL). Se encarga de enlazar el hardware del dispositivo con las aplicaciones y el software.

Linux proviene del Sistema Operativo Unix (de los años sesenta), desarrollado en de los Laboratorios Telefónicos “Bell”. Se fusionó con el sistema GNU en los años ochenta, cuando era aún un proyecto y necesitaba un núcleo capaz de controlar el hardware. Así, el estudiante Linus Torvalds completó el sistema GNU con el núcleo Linux y denominó el conjunto del sistema con el núcleo libre como GNU/Linux.



Es decir, por GNU/Linux se entiende la unión del núcleo o “Kernel” libre (proveniente de Unix) Linux con el Sistema Operativo GNU. Cualquier desarrollador tiene la posibilidad de utilizar este sistema según la Licencia Pública General de GNU debido a que está completamente desarrollado como software libre.

Para el desarrollo de esta aplicación se ha utilizado el protocolo Bluetooth, que no se diferencia demasiado de otros protocolos de comunicación como el TCP (“Transmission Control Protocol”). Para que se realice una conexión Bluetooth es necesaria la siguiente secuencia de pasos:

1. El dispositivo B se configura como servidor para permitir a otros conectarse con él.
2. El dispositivo A debe solicitar acceso de conexión al dispositivo B. Por lo tanto, debe ser conocida la dirección de los dispositivos, sino, no será posible el enlace.

El funcionamiento, de forma más detallada, sería de la siguiente forma:

1. Se crea en el servidor un socket de tipo “BluetoothServerSocket” para que escuche peticiones de conexión.
2. El servidor debe permanecer en espera ocupada a solicitudes de conexión (se utiliza el método “accept()”, que bloquea el hilo hasta recibir la conexión).
3. A partir de la dirección del dispositivo del cliente se instancia en éste un objeto “BluetoothDevice”.
4. A partir de este objeto, el cliente crea un socket “BluetoothSocket”.
5. A través del socket realiza una solicitud de conexión.
6. El servidor acepta la conexión, abre un nuevo “BluetoothSocket” y se lo comunica al cliente (lo que indica que la conexión ha sido realizada con éxito).
7. Se envían los datos a la interfaz a través de un “Handler”, mediante la escritura en el flujo de salida.

En la siguiente imagen, figura 24, se puede ver un diagrama del protocolo de conexión por bluetooth:

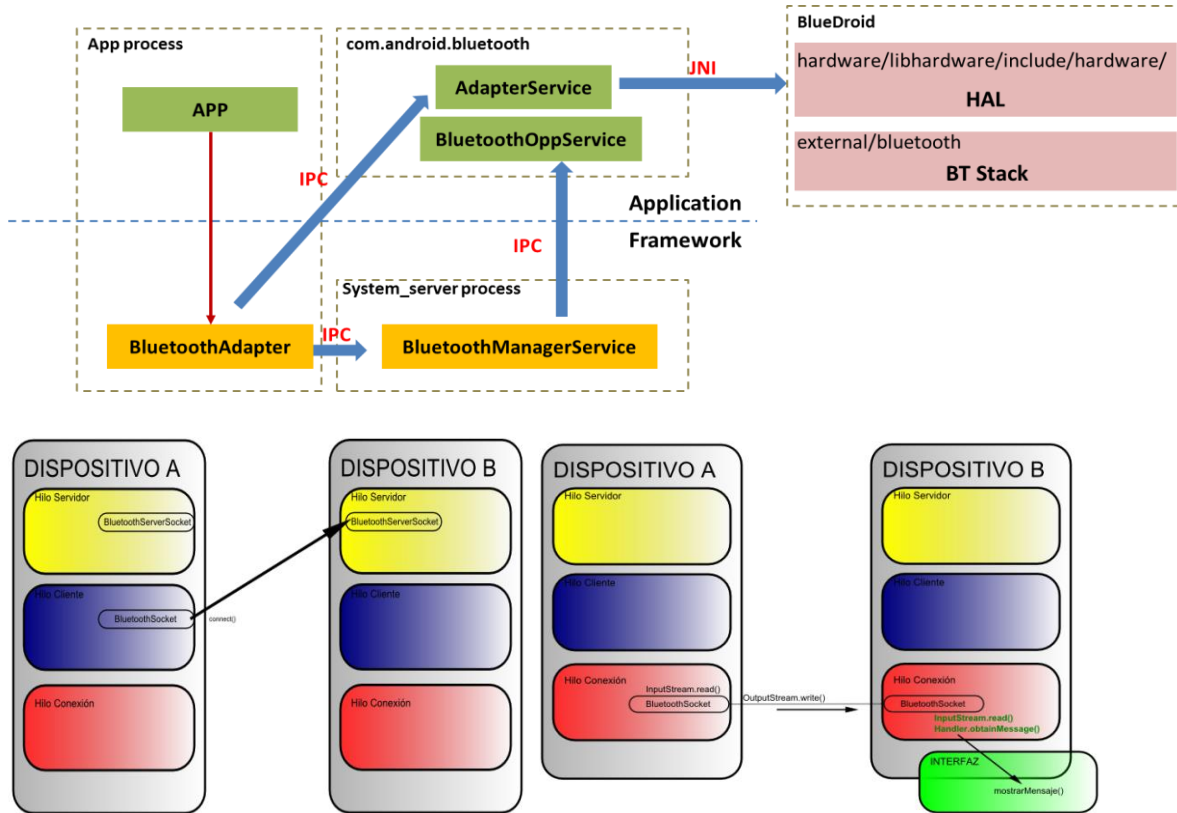


Figura 24: Diagrama de conexión Bluetooth.

Como requisito del proyecto se ha añadido un reconocedor de voz para dar al usuario la capacidad de desplazarse por la aplicación de forma más dinámica sin utilizar la interfaz táctil. Este sistema, permite comparar las órdenes dadas por el individuo y comparar la indicación con las palabras almacenadas en el sistema, en función de la tarea indicada para cada una, el sistema llevará a cabo una acción u otra.

Por definición, se considera un sistema de reconocimiento de voz a “una herramienta computacional capaz de procesar la señal de voz emitida por el ser humano y reconocer la información contenida en ésta, convirtiéndola en texto o emitiendo órdenes que actúan sobre un proceso.”

En la imagen mostrada a continuación, se puede ver un diagrama de funcionamiento del proceso de reconocimiento, mediante el cual la aplicación capta las palabras del usuario y las transforma en texto:

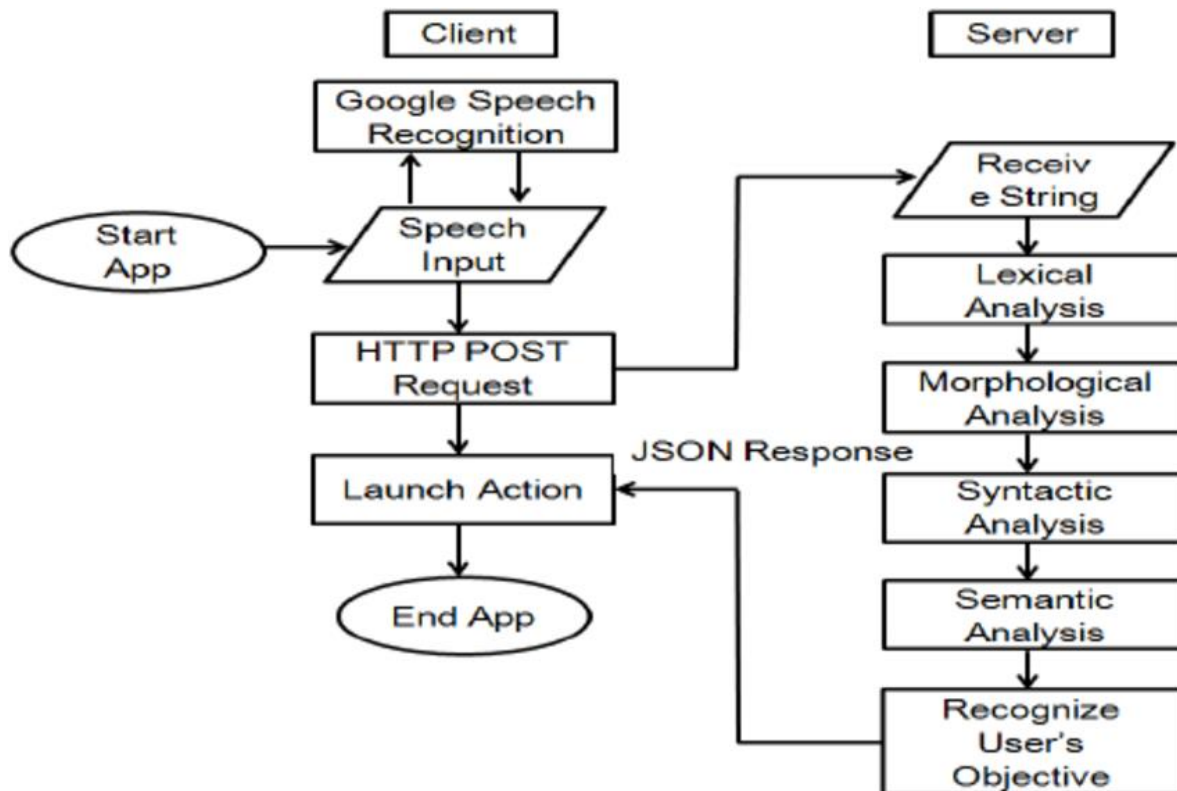


Figura 25: Diagrama reconocimiento de voz Android.

Los sistemas de reconocimiento de voz pueden clasificarse según 4 criterios:

- Entrenabilidad: puede que el sistema necesite algún tipo de entrenamiento previo.
- Dependencia del hablante: el sistema puede o no depender del habla de cada usuario.
- Continuidad: el sistema puede entender un habla continua o tener la necesidad que el usuario se detenga cada cierto número de palabras o tiempo.
- Robustez: indica la capacidad del sistema a funcionar con ruido o problemas vocales, su capacidad de entendimiento.
- Tamaño del dominio: depende de la cantidad de palabras que puede gestionar.



Las aplicaciones más usuales en las que se utiliza el reconocimiento de voz se pueden agrupar en:

- Dictado automático: Es la más común, transforma lo dicho por el usuario en texto.
- Control por comandos: sistemas diseñados para ejecutar órdenes, reconocen un vocabulario muy reducido, lo que mejora el rendimiento de la aplicación.
- Telefonía: sistemas PBX que permiten a los usuarios ejecutar comandos para navegar por un menú.
- Sistemas portátiles: para dispositivos de pequeñas dimensiones, como teléfonos o relojes.

En esta aplicación se ha utilizado un reconocimiento por control por comandos, dado que el objetivo es ordenar ejecuciones con un vocabulario establecido y permitir a usuarios discapacitados prescindir de los botones lo máximo que sea posible.

La mejora tanto en usabilidad como en robustez de este sistema permitirá mejorar la experiencia con el sistema aquí establecido (con una mayor reserva de palabras y un mejor entendimiento del lenguaje por parte del sistema).

En la siguiente imagen (figura 26), se puede ver el icono de la aplicación desarrollada por Google para el sistema Android y de cuya implementación (en código abierto) se ha basado la funcionalidad de reconocimiento utilizada en este trabajo.

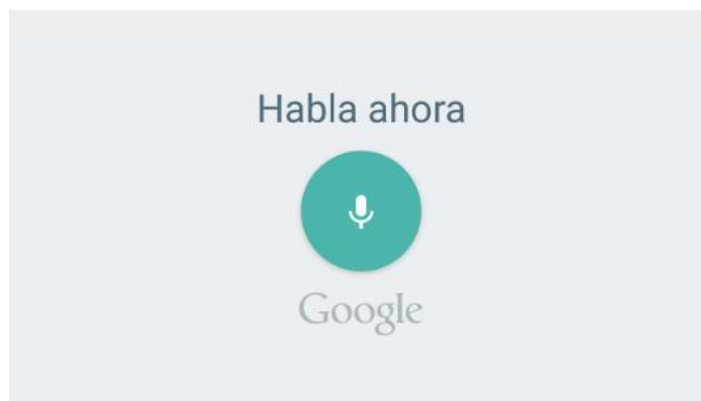


Figura 26: Icono “Habla ahora” de Google.

3.3.2 Descripción general del sistema

La aplicación comienza mostrando las instrucciones de uso. El usuario puede desplazarse a izquierda o derecha y cuando desee comenzar debe pulsar el botón “X”:

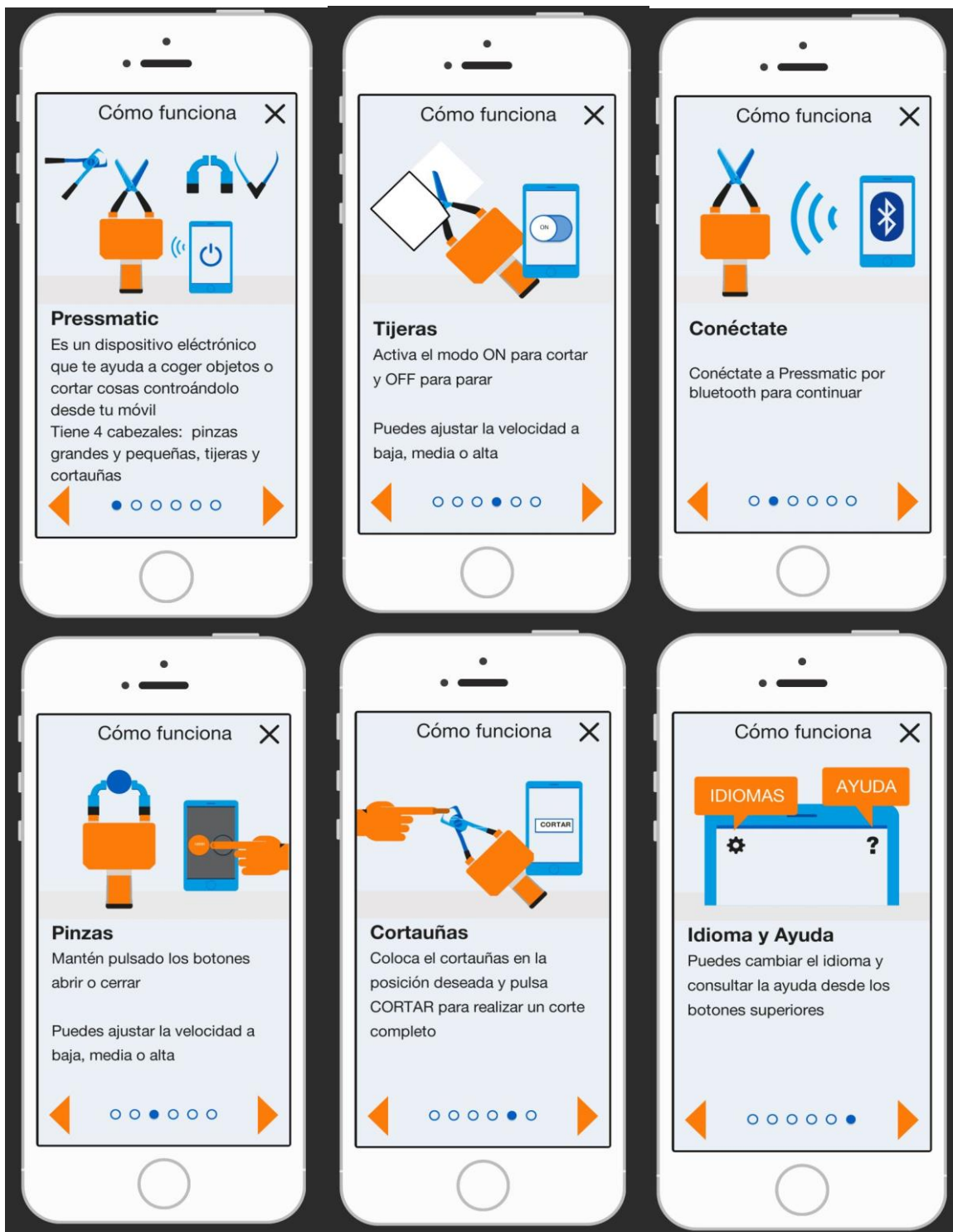


Figura 27: Instrucciones Pressmatic



Seguidamente, si no está habilitado el Bluetooth del terminal, aparecerá una pantalla con un mensaje de aviso de la necesidad de hacerlo, con dos botones para elegir activarlo o no (imagen de la izquierda). Si se elige “no”, al intentar conectar con Pressmatic (imagen de la derecha) volverá a indicar la necesidad de habilitar el Bluetooth y entrará en un bucle hasta que el usuario decida o activarlo manualmente o indicar “sí”.

Desde estas actividades, es posible cambiar el idioma de la aplicación (utilizando el icono de la rueda), o volver al menú de ayuda (con el icono de interrogación).



Figura 28: Actividades Bluetooth: Izquierda: habilitar. Derecha: conectar.

Tras conectar la aplicación con el dispositivo Pressmatic, el usuario ha de elegir una herramienta acorde con el utensilio colocado físicamente en el aparato.

El usuario puede moverse de izquierda a derecha buscando el elemento que necesita, además de tener la capacidad de cambiar el idioma y de volver a consultar las instrucciones.

Tras seleccionar un utensilio, aparecerá una nueva pantalla con la acción a realizar con cada uno de ellos. En todas ellas, existe la posibilidad de modificar la velocidad de acción en cada orden enviada desde el terminal al dispositivo.

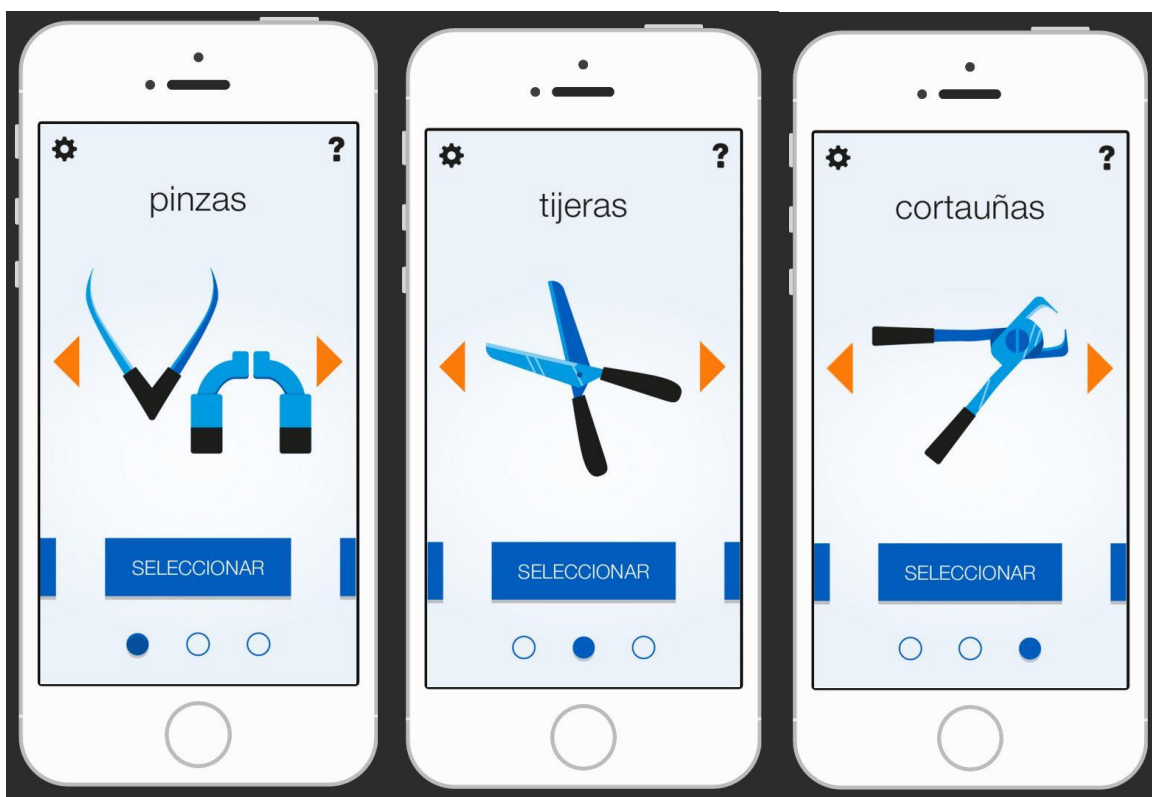


Figura 29: Herramientas disponibles.

En todo momento, el usuario deberá vigilar la calidad de la conexión entre el terminal y el dispositivo, si la conectividad falla, volverá a la pantalla de habilitar de nuevo el bluetooth y conectar a Pressmatic.



3.3.3 Descripción de componentes

Los principales bloques que proveen de funcionalidad la aplicación y son la base del proyecto los forman: el reconocimiento de voz, el diseño de la interfaz, el cambio de idioma y las órdenes mediante Bluetooth.

3.3.3.1 Reconocimiento de voz

Para llevar a cabo el reconocimiento de voz se ha utilizado una de las características del API (*Application program interface*) de Android “*speech recognition*”, una interfaz de desarrollo que permite a desarrolladores terceros acceder a funcionalidades de reconocimiento de voz y procesamiento de lenguaje natural.

Para poder utilizar este reconocedor de forma off line es necesario habilitar esta opción en los ajustes del terminal donde se esté ejecutando la aplicación:

en Ajustes->Idioma y entrada->Teclado y métodos de entrada->Lector de voz de Google

Es necesario utilizar una función para comprobar si el dispositivo tiene acceso a internet, pues sino la aplicación de reconocimiento no podrá trabajar. Si el terminal dispone de conexión, entonces la intención (“Intent”) de reconocer el habla se abrirá y comenzará la escucha; si el reconocimiento es verificado, el sistema responderá con un “ok”.

El proceso se basa en la llamada al método “startActivityForResult()” utilizando la acción “ACTION_RECOGNIZE_SPEECH”, lo que dará comienzo a la actividad de lectura de voz cuyos resultados podrán gestionarse en “onActivityResult()”.

Antes de capturar el mensaje del usuario es necesario comprobar si el reconocimiento está habilitado, si no lo está aparecerá un mensaje de aviso, si lo está entonces la aplicación instanciará un “RecognizerIntent” para pasarle parámetros.

Una vez capturado el mensaje se gestiona en el método “onActivityResult” y se lleva a cabo la acción requerida.



3.3.3.2 Diseño de la interfaz

Para el desarrollo de la interfaz de usuario se ha seguido el diseño gráfico elaborado en un trabajo previo, basado en la experiencia de usuario [1]. La implementación gráfica se divide en varias pantallas que engloban los bloques que componen la aplicación:

1. Instrucciones de uso de la aplicación
2. Conectividad
3. Idioma
4. Selección de herramienta

Buscando cumplir dichas condiciones y ajustar la experiencia a las exigencias impuestas, ha sido necesario instanciar botones de forma circular, botones enmarcados, vistas que aparecen y desaparecen, modificación del idioma de la aplicación y búsqueda de colores para ajustar el diseño a la versión ideal.

Con el fin de crear los materiales deseados (botones con formas y estructuras fuera de lo común), se han utilizado imágenes como botones, añadido fronteras para formar contorno en el borde de botones rectangulares y de ficheros “drawable” .xml con selectores donde se crea la forma circular de los botones, posteriormente añadida a los “layouts” de la aplicación.

En la creación de los mensajes de alerta cuando el bluetooth del terminal no está activado, se ha utilizado la capacidad de los “layout” de Android de hacerse visibles o invisibles según la condición deseada (en este caso que el valor del adaptador bluetooth sea nulo).

La forma visual y diseño artístico de la aplicación han sido dados de antemano según los requisitos buscados. En la siguiente figura (figura 30) se muestra la pantalla de inicio de la aplicación:



Figura 30: Pantalla de inicio.

3.3.3.3 Modificación del lenguaje

Hoy en día, cualquier aplicación de teléfono está disponible en varias lenguas o permite el cambio del idioma una vez ejecutada la aplicación. Esto se debe a la necesidad de los desarrolladores de internacionalizar su producto para facilitar y mejorar su posterior entrada en el mercado y venta.

Debido a esto, Android Studio dispone de varios métodos para el cambio de idioma, desde la traducción automática provista por un software externo hasta la traducción manual por parte del programador, mediante la creación de un segundo fichero con la traducción del texto.

Para llevar a cabo el cambio de idioma en la interfaz (español-inglés) ha sido necesaria la creación de un segundo valor “string.xml” y localizarlo según el lenguaje buscado (en este caso, “string-en”), donde se encuentran traducidas las entradas de texto (manualmente). Así, mediante una llamada a la clase “Locale” de Android es posible modificar el idioma de la aplicación cuando el usuario lo desee.



3.3.3.4 Órdenes enviadas mediante Bluetooth

Aunque el proceso de comunicación entre la aplicación y el dispositivo Pressmatic se ha basado en el ya implementado en el anterior proyecto, respetando su estructura de mensajes mediante sockets y las condiciones de funcionamiento, se ha adaptado dicho protocolo al nuevo diseño de la aplicación, eludiendo ciertas órdenes de funcionamiento y reestructurando las clases y los métodos utilizados antes en una sola actividad.

El protocolo de comunicación por Bluetooth, que se puede estructurar en función de las necesidades del programador, en esta aplicación se forma con tres hilos: uno que escucha las conexiones entrantes, otro que se conecta con el dispositivo y el que realiza la transmisión de datos una vez hecha la conexión.

Para saber qué herramienta ha elegido por el usuario se ha escogido el método de intercambio de mensaje a través de las actividades “putExtra”. De esta forma, al elegir un utensilio, éste se etiqueta con un atributo con una sigla que es enviada a la actividad donde se lleva a cabo el envío de datos con el dispositivo Pressmatic que asocia ese elemento elegido con el que tiene encajado en la parte superior de su tronco.

Al llevar a cabo las acciones se utiliza el traspaso de datos mediante bluetooth por sockets. Cuando el usuario desea llevar a cabo una tarea, una vez elegida la herramienta de trabajo, se envía la acción a realizar en por un socket a la clase que implementa la comunicación Bluetooth.

Esta orden se identifica mediante un número entero, que varía según la velocidad elegida, sistema que permite controlar al mismo tiempo el tipo de acción y la velocidad a la que ésta se realiza.



Capítulo 4



Experimentación

El proceso de experimentación de este proyecto se ha dividido en dos fases: una de simulación, en la que se han utilizado simuladores y terminales Android con diferentes versiones del sistema operativo; y otra de pruebas, en la que se ha verificado la funcionalidad de la aplicación con el dispositivo asistencial Pressmatic en diferentes situaciones y contextos.

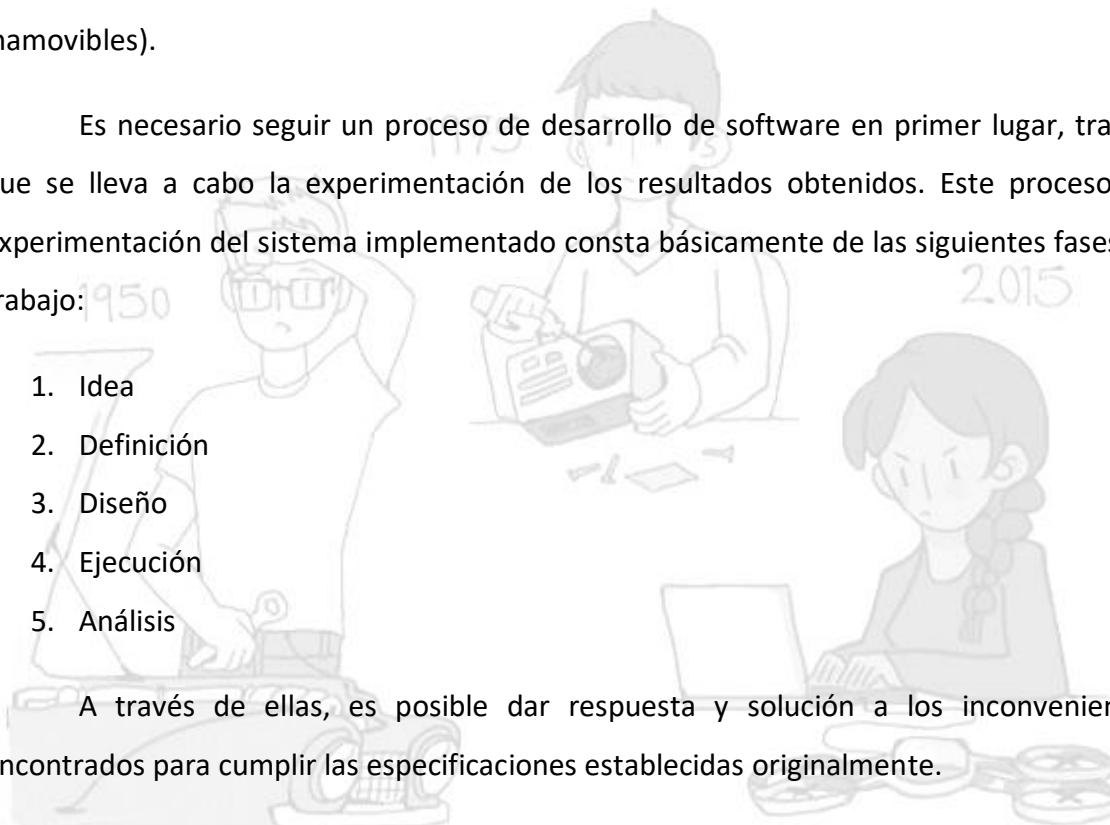
La experimentación tiene como objetivo el reconocimiento de los motivos por los cuales se obtienen los resultados, ya sean positivos (según los requisitos expuestos) o erróneos (con problemas en la ejecución). Así, la fortaleza de la experimentación radica en el entendimiento del proceso y sus consecuencias para poder llevar a cabo las modificaciones o manipulaciones necesarias para alcanzar los resultados deseados.

La ingeniería de software (IS) es una disciplina, aún con poco recorrido, en la cual se construyen sistemas que no están basados en leyes absolutas o inmutables (al contrario que otras ingenierías, establecidas a partir de leyes físicas establecidas como inamovibles).

Es necesario seguir un proceso de desarrollo de software en primer lugar, tras lo que se lleva a cabo la experimentación de los resultados obtenidos. Este proceso de experimentación del sistema implementado consta básicamente de las siguientes fases de trabajo:

1. Idea
2. Definición
3. Diseño
4. Ejecución
5. Análisis

A través de ellas, es posible dar respuesta y solución a los inconvenientes encontrados para cumplir las especificaciones establecidas originalmente.





4.1 Fase 1: Simulaciones

En el comienzo del desarrollo de la aplicación, se ha utilizado el emulador de Android Studio para visualizar el aspecto de las imágenes, colores y botones especificados en los requerimientos del diseño.

Al realizar pruebas de activación, aviso y conexión de Bluetooth, fue necesario el uso de terminales reales, con diferentes versiones en los sistemas operativos Android para comprobar la efectividad de las condiciones de aviso de activación, búsqueda de dispositivos y ajuste del diseño de las actividades (ancho y largo).

- Emulador Android Studio: permite comprobar los diseños, calidad de las funcionalidades y comprobación de los condicionales de la implementación (como las alertas para habilitar el bluetooth).
- Terminales reales: permiten acceder a la red bluetooth, ajustan el diseño a la pantalla, disponen de la opción para probar el sintetizador...

Los mayores inconvenientes se dieron por la tecnología Bluetooth, ya que, al cambiar el diseño de la aplicación, ha sido necesario reestructurar la implementación prescindiendo de algunas clases y combinando otras para hacer una aplicación más sencilla de entender y mejor organizada.

Para comprobar el funcionamiento del reconocimiento de voz también era necesario el uso de terminales reales. Este proceso fue rápido y sin apenas inconvenientes, la mayoría encaminados a la calidad del sintetizados de habla.

Tras la verificación del funcionamiento del código implementado en la aplicación se pasó a la fase 2, en la que se estuvo trabajando directamente sobre el dispositivo real y se comprobaron las órdenes de ejecución de tareas y las pruebas de conexión con el aparato.



4.2 Fase 2: Pruebas con el prototipo real

La aplicación ha sido testada por el tutor de este proyecto y el director del proyecto Pressmatic original. En ellas se ha comprobado tanto el correcto funcionamiento del software desarrollado como la utilidad del diseño a la hora de mejorar la interfaz de la aplicación facilitando la experiencia del usuario.

El proceso de prueba se ha dividido en una serie de ensayos que han comprobado los requisitos exigidos en el trabajo. Éstos debían responder satisfactoriamente a una serie de preguntas:

- ¿Es posible conectar la aplicación con el dispositivo Pressmatic?
- Al cancelar la conexión, ¿es posible volver a conectar los dispositivos?
- Tras la conexión, ¿responde a las órdenes indicadas desde el terminal?
- Si se inhabilita el protocolo bluetooth del teléfono, ¿se desconecta automáticamente el dispositivo y aparece un mensaje de alerta? De ser así, ¿puede volver a conectarlo de nuevo sin problemas?
- ¿Funciona correctamente el reconocimiento de voz para transmitir órdenes a Pressmatic?
- ¿Influye el idioma elegido (tanto oral como en texto) en el correcto funcionamiento de la aplicación?
- ¿Ha sido satisfactoria su experiencia con el nuevo diseño de la aplicación?, ¿qué puntos mejoraría y cuáles han mejorado su visión respecto a la versión original?
- ¿Qué funcionalidades le gustaría añadir o recuperar del anterior proyecto?
- ¿Adquiriría esta aplicación del mercado de aplicaciones de Android?, ¿solo gratis o a un precio asequible?



4.2.1 Primera sesión de pruebas

En la primera prueba realizada con el prototipo real se encontraron problemas a la hora de conectar el terminal con el dispositivo Pressmatic.

El objetivo era comprobar la repuesta del sistema a las preguntas anteriormente formuladas:

1. ¿Es posible conectar la aplicación con el dispositivo Pressmatic?
 - a. Para poder conectar la aplicación con Pressmatic, era necesario que el dispositivo Pressmatic estuviese encendido previamente y con el bluetooth activado antes de ejecutar la aplicación, sino, al intentar conectar con el dispositivo aparecía un mensaje de error: “No ha sido posible realizar la conexión”.
 - b. La aplicación detectaba mediante bluetooth la presencia del dispositivo, aunque se deshabilitara el bluetooth antes o después de conectarse y se habilitara de nuevo.
2. Al cancelar la conexión, ¿es posible volver a conectar los dispositivos?
 - a. Sólo si se detenía la aplicación y era ejecutada de nuevo, sino, se obtenía un objeto nulo por parte del dispositivo a la hora de intentar establecer la conexión bluetooth, con el mensaje: “Conexión rechazada”.
3. Tras la conexión, ¿responde a las órdenes indicadas desde el terminal?
 - a. No. Éste fue el principal inconveniente encontrado en esta primera prueba, aunque era posible conectar ambos dispositivos, la aplicación parecía no recibir un objeto válido por parte del dispositivo para confirmar la conexión.
 - b. Se inhabilitó por completo el botón “X” que lleva las actividades secundarias (instrucciones, cambio de lenguaje, aviso de conexión bluetooth...) a la actividad principal Pressmatic.java donde se almacenan los



- comparativos para activar el menú de herramientas y las órdenes para ejecutar las acciones del dispositivo.
- c. No fue posible seleccionar una herramienta en el terminal pesa a que el dispositivo estaba conectado con la aplicación, ya que en ningún momento apareció la pantalla de selección de utensilios. Se encontró un error en la clase principal que no adquiría y comparaba el objeto con la información de la conexión proveniente de Pressmatic.
 - d. No fue posible enviar órdenes desde la aplicación al dispositivo en esta prueba.
4. Si se inhabilita el protocolo bluetooth del teléfono, ¿se desconecta automáticamente el dispositivo y aparece un mensaje de alerta? De ser así, ¿puede volver a conectarlo de nuevo sin problemas?
- a. Efectivamente, al desconectar el bluetooth del terminal o del dispositivo aparece en el teléfono el mensaje “Se ha perdido la conexión”.
 - b. Sólo es posible conectarlo de nuevo si se detiene la ejecución de la aplicación y se vuelve a iniciar, habilitando primero el bluetooth del dispositivo Pressmatic.
5. ¿Funciona correctamente el reconocimiento de voz para transmitir órdenes a Pressmatic?
- a. No fue posible transmitir órdenes al dispositivo, pero la funcionalidad por voz ejecutaba la aplicación de forma similar a los botones.
6. ¿Influye el idioma elegido (tanto oral como en texto) en el correcto funcionamiento de la aplicación?
- a. No, la aplicación respondía con la misma ejecución en ambos idiomas.
 - b. Funcionalidad de idioma implementada correctamente.
7. ¿Ha sido satisfactoria su experiencia con el nuevo diseño de la aplicación?, ¿qué puntos mejoraría y cuáles han mejorado su visión respecto a la versión original?



- a. En un primer lugar fue necesario acudir al manual de instrucciones de uso presente en este documento, aun con las instrucciones presentes en la aplicación.
 - b. Es necesario clarificar las instrucciones para no necesitar una ayuda externa por parte del desarrollador o de un manual de uso.
8. ¿Qué funcionalidades le gustaría añadir o recuperar del anterior proyecto?
- a. Primero es necesario recuperar las funcionalidades básicas de conexión y ejecución de órdenes.
9. ¿Adquiriría esta aplicación del mercado de aplicaciones de Android?, ¿solo gratis o a un precio asequible?
- a. Para esta prueba la aplicación aún no está completamente desarrollada y lista para comercializarse.



4.2.2 Segunda sesión de pruebas

En este ensayo se pretendía comprobar los resultados fallados en el primero y tras comprobar los defectos de la aplicación. Las preguntas objetivo a responder eran:

1. ¿Es posible conectar la aplicación con el dispositivo Pressmatic?
 - a. Sí, pero no es posible conectar con Pressmatic si antes de ejecutar la aplicación no se ha enlazado con el bluetooth del teléfono y se almacena como dispositivo conocido.
2. Al cancelar la conexión, ¿es posible volver a conectar los dispositivos?
 - a. No, al cancelar la conexión no aparece la lista de dispositivos bluetooth disponibles de nuevo. Es necesario detener la aplicación y ejecutarla de nuevo.
3. Tras la conexión, ¿responde a las órdenes indicadas desde el terminal?
 - a. Al conectar con el aparato la aplicación accede al menú de herramientas, pero no es capaz de llegar a las pantallas de acción. Es necesario revisar el código para localizar las condiciones que impiden el acceso a las pantallas de órdenes.
4. Si se inhabilita el protocolo bluetooth del teléfono, ¿se desconecta automáticamente el dispositivo y aparece un mensaje de alerta? De ser así, ¿puede volver a conectarlo de nuevo sin problemas?
 - a. Al desconecta el bluetooth aparece un mensaje de aviso, pero la aplicación no vuelve a la pantalla inicial de conexión y tampoco es posible conectar de nuevo a Pressmatic sin salir de la aplicación y ejecutarla de nuevo.
5. ¿Funciona correctamente el reconocimiento de voz para transmitir órdenes a Pressmatic?
 - a. Se encuentran problemas respecto al idioma del reconocedor. Sólo reconoce el idioma que esté establecido en la aplicación, pero si en el terminal hay un idioma distinto parece tener problemas para captar las palabras.



4.2.3 Tercera sesión de pruebas

En este tercer proceso, se pretendió dar respuesta a los problemas encontrados tanto en la conexión como para ejecutar órdenes. También se depuraron defectos en el diseño:

1. ¿Es posible conectar la aplicación libremente, ejecutar las órdenes y controlar el estado de la conexión en todo momento?
 - a. La aplicación ya es capaz de conectarse sin enlazar el dispositivo Pressmatic con el bluetooth del teléfono, aunque encuentra algunos problemas para conectar de nuevo si se pierde la conexión en medio de la ejecución de la aplicación (en la pantalla para conectar con el aparato).
 - b. Están disponibles los menús de herramientas y acciones para trabajar con el aparato, pero se encuentran inconvenientes de ejecución para el control de las pinzas.
2. ¿Cuáles son los inconvenientes encontrados a la hora de ejecutar las acciones?
 - a. En algunas ocasiones se pierde la conexión de forma inesperada, y es necesario reiniciar el bluetooth del teléfono o el dispositivo Pressmatic. Al perder la conexión en medio de la ejecución de una acción, ésta no se detiene y es necesario utilizar la interfaz del dispositivo para controlarla.
 - b. No es posible mantener pulsados los botones para llevar a cabo una ejecución continua hasta dejar de presionar.
 - c. Tras conectar con el dispositivo, si se pretende cambiar el idioma o acudir a la ayuda no es posible volver después al menú de herramientas. Es necesario reiniciar la conexión con Pressmatic.
3. ¿Coincide el diseño en funcionalidad y estilo con los requisitos del trabajo?
¿Responde a las pruebas de usuario?



- a. Es necesario el cambio en el tipo de algunos botones para adaptar la funcionalidad a lo que se requiere (por ejemplo, una ejecución continua al pulsar un botón y que se detenga al pulsarlo de nuevo).
 - b. Se depuran pequeños defectos de idioma, color y se pretende iluminar un botón con cuando esté en proceso de ejecución (elección de velocidades).
4. ¿Cómo encuentra el estado de avance de la aplicación?
- a. Hay que mejorar la comprobación del estado de la conexión en todo momento, y permitir al usuario navegar por los distintos bloques de la aplicación en cualquier momento.
 - b. Se aprecian avances desde el comienzo de las pruebas, tanto en las funcionalidades de la aplicación como en su nivel de usabilidad por parte del usuario.



4.2.4 Cuarta sesión de pruebas y posteriores mejoras

Tras los primeros avances e inconvenientes encontrados en el desarrollo de la aplicación, para esta prueba se pretende recuperar antiguas funcionalidades (como la capacidad de cambiar el tamaño de la letra y su tipo), implementar el pulsado continuo de un botón y cambiar el estilo de cambio de pantalla por “SlideView” (para flexibilizar el uso de botones en la aplicación).

1. Aparecen problemas para implementar un botón que permita el pulsado continuo mientras se envían órdenes a Pressmatic. El dispositivo Pressmatic sólo acepta la primera orden y no se detiene al levantar el dedo.
2. Se necesita implementar un sistema de comunicación entre el terminal y el dispositivo Pressmatic para indicar en qué módulo de la aplicación se encuentra el usuario en cada momento para que se corresponda con lo que se muestra en la pantalla de Pressmatic.
3. Aunque se encuentra implementada la opción, el terminal no responde a la hora de cambiar el tamaño y tipo de letra.
4. Se comienza a cambiar la estructura de la aplicación para permitir al usuario navegar entre pantallas arrastrándolas con el dedo. Ya no es necesario utilizar los botones izquierda y derecha.
5. También, como implementaciones adicionales, se pretende utilizar un mensaje de transcripción del reconocimiento de comandos del usuario. Además de un botón que indique el estado en el que se encuentra la conexión en todas las actividades.

Como la aplicación se ha estructurado con una actividad por pantalla de la aplicación, y en cada una de ellas su correspondiente funcionalidad (botones, direccionamiento a la actividad correspondiente...), el desarrollo de la interfaz con SlideView para el cambio en las pantallas de usuario deja de parecer viable debido a que requeriría la modificación de toda la estructura en el diseño de la aplicación, por lo que se pospondrá para futuras mejoras (al igual que el uso de fragments para pantallas como la conexión bluetooth).



Finalmente, se consigue implementar dos botones para controlar las pinzas según el pulsado continuado de los mismos. Es decir, al mantener uno de los botones pulsando comienza la acción correspondiente y al levantar el dedo, se detiene la tarea.

Esto ha sido posible gracias al conocimiento de las órdenes que controlan Pressmatic que pueden ser enviadas a través del hilo de comunicación.

Los principales problemas encontrados se han dado con el gobierno del estado de la conexión en todas las actividades, sin perder el estado de comprobación ni interferir en los valores de la variable de control de una actividad a otra. Esta tarea ha sido, sin duda, la más compleja.

Tras varias pruebas y estudio del módulo de control de la conexión Handler, se ha avanzado en este apartado. Con la idea de obtener el estado de la conexión y mantener una variable entre todas las actividades de la aplicación, se ha intentado utilizar una variable global que cambia su valor según lo comunicado por el hilo que conecta el terminal con Pressmatic.

El uso de esta variable encontró problemas al cambiar de valor en cada módulo Handler de cada una de las actividades, por lo que, posteriormente, se utilizó la variable de forma privada en cada actividad, pero al testear la aplicación aparecían incoherencias en el sentido correcto de la ejecución.

Es necesario mejorar e implementar diferentes tipos de estructuras para revisar la conexión entre el terminal y el dispositivo. Las clases AsyncTask podrían aportar progresos en el envío y recibo de mensajes, su escucha continuada y el intercambio de variables entre actividades para controlar su valor.



4.2.4.1 Evolución respecto a la aplicación original

Durante la construcción de la aplicación se han logrado mejoras tanto a nivel de usabilidad como de funcionalidad. El progreso en cuanto a funcionalidad se basa en cuatro puntos que han permitido el desarrollo de la interfaz para el usuario:

1. El mantenimiento de la conexión con el dispositivo Pressmatic una vez establecida, la capacidad de monitorizar el estado de dicha conexión (conocer el estado de la misma, si permanece o se pierde en todas las actividades) y de volver a constituir el enlace si se ha perdido o cancelado sin necesidad de reiniciar la aplicación.
2. La implementación de un reconocedor de voz, que permite llevar a cabo acciones en la aplicación y enviar órdenes a Pressmatic, a través de la verificación de los comandos por voz registrados por el usuario.
3. El desarrollo de un sistema multilenguaje, que permite al usuario cambiar el idioma de la aplicación a español o inglés desde cualquier módulo sin afectar el estado de la conexión (que se mantiene en un hilo en segundo plano).
4. El progreso en la funcionalidad de los botones, como el pulsado continuo en la actividad para controlar las pinzas, que se abren o cierran según se mantenga presionado el botón correspondiente. También un botón para salir de la aplicación o para mostrar vistas ocultas.

Al añadir estas nuevas capacidades, se ha logrado la mejora de la interfaz dividiendo los módulos de la aplicación en varias actividades por las que puede desplazarse el usuario. Además de proporcionar un diseño más atractivo.

A continuación, se muestra en resumen la evolución de la interfaz que se ha trabajado en este proyecto, comparando cada uno de los módulos con la aplicación base de la que se ha partido mediante unas capturas de pantalla (en el terminal de trabajo).

La imagen de la izquierda pertenece a la aplicación antigua, la de la derecha a la desarrollada en este trabajo:

IMPLEMENTACIÓN DE APLICACIÓN ANDROID PARA CONTROL DE DISPOSITIVO ASISTENCIAL PRESSMATIC



Figura 31: Progreso del diseño aplicación Pressmatic





Capítulo 5



Gestión del proyecto

Gestionar un proyecto quiere decir planificar y orientar los procesos que se llevan a cabo en el mismo de manera metódica de principio a fin. Según el Instituto de Gestión de Proyectos, los procesos siguen cinco etapas de desarrollo:

1. Iniciación en los temas a tratar
2. Planificación
3. Ejecución
4. Control
5. Cierre

Existen varias perspectivas para llevar a cabo la gestión de las diferentes actividades que conforman un proyecto, éstas se pueden incluir en: enfoque lean (producción esbelta), reiterativo, incremental y en fase. Para este trabajo, se ha seguido el método tradicional: inicio, diseño, ejecución, monitorización y cierre.

Este trabajo ha sido desarrollado en 4 meses continuos, desde la preparación de los conceptos teóricos hasta elaborar la memoria. Este desarrollo se puede dividir en cuatro etapas:

1. Estudio de las condiciones requeridas y aprendizaje del lenguaje.
2. Implementación de la aplicación.
3. Experimentación, prueba y mejora de los resultados.
4. Documentación: memoria y presentación.

Para especificar la estructura temporal y de organización en la que se ha desarrollado este trabajo, se ha incluido un diagrama de Gantt (herramienta gráfica para exponer la dedicación en una tarea) y una tabla explicativa.

Antes de comenzar con el desarrollo del trabajo, es necesario llevar a cabo un estudio sobre el reglamento y el entorno socio-económico presente.



5.1 Marco regulador

Según la legislación vigente [8] es necesario tener en cuenta la ley de protección de datos de carácter personal y la legislación de servicios de la sociedad de la información y de comercio electrónico. En el artículo primero de esta ley se indica que:

“Tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor e intimidad personal”.

La Ley 34/2002, de 11 de julio, de Servicios de la sociedad de la información y de comercio electrónico y la Ley 32/2003, de 3 de noviembre, General de Telecomunicaciones atribuyen competencias en materia sancionadora a la Agencia Española de Protección de Datos. Incluyen tanto personas físicas como jurídicas.

Si el objetivo de este trabajo fuese comercializar la aplicación, aunque al terminar esta memoria el desarrollo continuará, es necesario ajustarse a unos aspectos legales presentes en España en este momento:

- Funcionalidades: deben ser lícitas, clarificando sus capacidades y sin describir competencias que no incluye.
- Derechos propios y de terceros: las aplicaciones son obras multimedia, formadas por varios elementos que requieren de licencias de uso y comercialización. Estos derechos de acceso, rectificación, cancelación y oposición al tratamiento, según ha afirmado el Tribunal Constitucional en su sentencia número 292/2000, constituyen facultades del derecho fundamental a la protección de datos:

“Sirven a la capital función que desempeña este derecho fundamental: garantizar a la persona un poder de control sobre sus datos personales, lo que sólo es posible y efectivo imponiendo a terceros los mencionados deberes de hacer”.



- Uso por parte de menores: para menores de 14 años afecta la legislación en materia de consumidores y usuarios, protección de datos...

“Corresponderá al responsable del fichero o tratamiento articular los procedimientos que garanticen que se ha comprobado de modo efectivo la edad del menor y la autenticidad del consentimiento prestado en su caso, por los padres, tutores o representantes legales.”

- Privacidad: es necesario informar previamente en la propia aplicación y que el usuario/paciente tenga la opción de decidir su grado de privacidad. Aunque no es incompatible el uso de datos que afecten a la privacidad (apartado 3) si el tratamiento de los datos tiene fines científicos.

Según Lo dispuesto en la Ley 12/1989, de 9 de mayo, Reguladora de la función estadística pública, la Ley 16/1985, de 25 junio, del Patrimonio histórico español y la Ley 13/1986, de 14 de abril de Fomento y coordinación general de la investigación científica y técnica, y sus respectivas disposiciones de desarrollo, así como a la normativa autonómica en estas materias.

- Licencia de uso y condiciones legales: el usuario/paciente deberá aceptarlas previamente.
- Publicidad: es la principal fuente de ingresos de las aplicaciones, debiendo estar debidamente indicada. Para ello, es necesario informar al interesado del tratamiento de datos susceptibles:

“Hayan sido facilitados por los propios interesados u obtenidos con su consentimiento para finalidades determinadas, explícitas y legítimas relacionadas con la actividad de publicidad o prospección comercial, habiéndose informado a los interesados sobre los sectores específicos y concretos de actividad respecto de los que podrá recibir información o publicidad.”



5.2 Entorno socio-económico

El objetivo principal de esta aplicación es prestar apoyo al paciente con las tareas que se pretenden realizar con el dispositivo Pressmatic. Su instalación y uso serán gratuitas previa adquisición del dispositivo, por lo que su impacto económico está supeditado al coste del aparato asistencial Pressmatic.

Las Tecnologías de Información y Comunicación (TIC) se han asentado y continúan revolucionando la sociedad actual. La implementación de software y hardware en áreas como educación, salud y seguridad contribuyen a la mejora de la calidad de vida de las personas y al progreso tecnológico.

En cuanto a materia asistencial, el uso de tecnologías avanzadas tiene un impacto positivo desde el punto de vista tanto del paciente como del sanitario, mejorando la rehabilitación de los pacientes y facilitando la labor del sanitario y los familiares responsables.

La necesidad de disponer de un terminal o tableta con sistema operativo Android y una versión lo bastante actualizada como para ejecutar la aplicación son los principales inconvenientes para este proyecto.

El objetivo de los teléfonos inteligentes fue en su origen que el usuario tuviese acceso a la red para facilitar su vida cotidiana, aunque en la actualidad numerosos estudios demuestran una gran dependencia desarrollada por las personas al usar este dispositivo.

También ha de tenerse en cuenta que la proliferación de tecnologías asistenciales puede llevar a la deshumanización de los tratamientos de rehabilitación. Este es un punto ético presente en la mayoría de los avances tecnológicos, pues desde la aparición de los primeros terminales y el avance de la tecnología ha producido la deshumanización tanto en las áreas de trabajo industriales como la disminución en las relaciones personales.



“Humanizar la salud, constituye un compromiso ético de considerar a la persona en su globalidad. Y una intervención holística, global, integral, necesita de una particular capacitación de los agentes sociales en el ámbito de la inteligencia del corazón, de las habilidades para entrar en el mundo personal y particular de la persona a la que se quiere acompañar, identificar y movilizar en ella no sólo los problemas y necesidades, sino el mundo de los significados, los recursos, las habilidades y los valores que pueden permitirle trabajarse a sí mismo y ser el mayor protagonista del proceso. Promover la dignidad intrínseca de todo ser humano constituye el fundamento último de toda acción humanizada” [21].

El impacto ambiental de las telecomunicaciones se refiere a la generación de residuos sólidos, electro smog, incremento de los niveles de ruido, los cambios en el uso del suelo e impacto visual [10].

Los residuos que generan los dispositivos electrónicos son conocidos como Residuos de Aparatos Eléctricos y Electrónicos (RAEE). Un teléfono móvil pesa entre 120-150 g con batería y puede llegar a generar 21,1 kg de residuos.

Según la Directiva Europea sobre la Restricción del Uso de Ciertas Sustancias Peligrosas en equipos eléctricos y electrónicos (RoHS). Materiales como el plomo, mercurio, cadmio, cromo hexavalente y los retardadores de llama bromados.

El Real Decreto 208/2005, de 25 de febrero, sobre aparatos eléctricos y electrónicos y la gestión de sus residuos, establece en su Disposición adicional primera la obligación de que todos los productores de aparatos eléctricos y electrónicos se inscriban en la Sección Especial del Registro de Establecimientos Industriales (REI), creado por Ley 21/1992, de 18 de julio.



5.3 Descripción de las fases del proyecto

Los proyectos de ingeniería se suelen dividir en cuatro pasos y un control, según el enfoque tradicional:

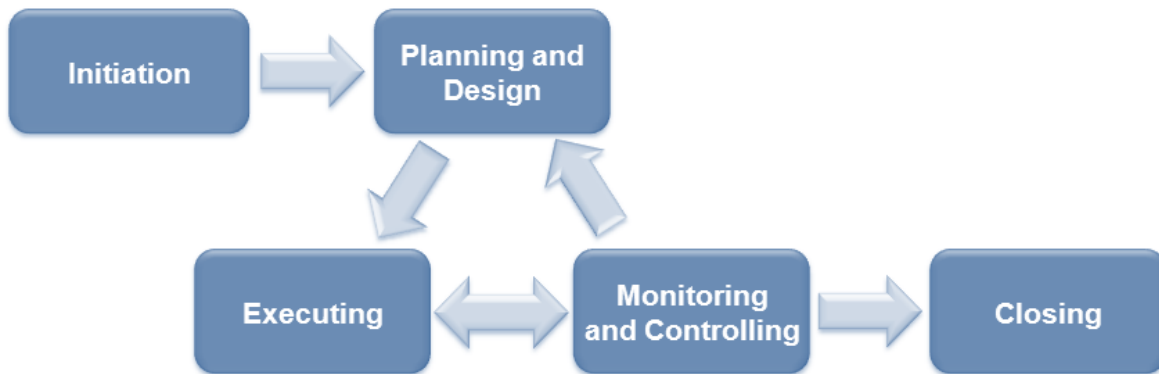


Figura 32: Fases de un proyecto.

Las fases de ejecución del proyecto se estructuran según la necesidad de cumplir los requerimientos exigidos en cada momento. Son cuatro puntos principales:

1. Estudio y aprendizaje del lenguaje de programación utilizado en la aplicación que se deseaba mejorar (en diseño y capacidades). Este proceso se enfocó en los objetivos a cumplir, es decir, se centró en la parte gráfica de la plataforma y en la adición de un sistema sintetizador de voz.
2. Desarrollo del software: en esta fase se llevó a cabo un estudio al detalle de la aplicación original, se modificaron y aumentaron clases y actividades y se implementaron los atributos solicitados.
3. Prueba de fallos: parte imprescindible en todo proceso, consistió en comprobar que los cambios realizados no afectaban al correcto funcionamiento del servicio. Además, se comprobó la usabilidad y marcha del reconocimiento de voz añadido.
4. Búsqueda de información y documentación del trabajo, desde la preparación de la información relacionada con el proyecto a la descripción del trabajo realizado.



5.4 Planificación

Para describir la planificación que se ha llevado a cabo se muestra a continuación una tabla (tabla 22) con la planificación del trabajo y una imagen con un diagrama de Gantt que indica el tiempo invertido en cada proceso. No se han contabilizado días de descanso ni vacaciones, además, el fin de semana se ha utilizado también como días de elaboración del trabajo.

En la tabla se presenta, en primer lugar, el nombre de la tarea realizada, después un desglose de los sub trabajos derivados de ella y a continuación, la fecha de inicio y fin de cada uno de los procesos, que en conjunto conforma la fecha de desarrollo de dicha tarea.

| Nombre de la tarea | Fecha de Inicio | Fecha final |
|----------------------------------|-----------------|-----------------|
| Estudio | 01-10-16 | 04-11-16 |
| Aprendizaje del lenguaje Android | 01-10-16 | 01-11-16 |
| Análisis de los requisitos | 02-11-16 | 04-11-16 |
| Implementación | 11-11-16 | 27-01-17 |
| Observación de la aplicación | 11-11-16 | 18-11-16 |
| Búsqueda de información | 18-11-16 | 02-12-16 |
| Desarrollo | 25-11-16 | 27-01-17 |
| Pruebas | 14-10-16 | 06-03-17 |
| Comprobación en terminal emulado | 14-10-16 | 16-11-16 |
| Ensayos en teléfonos reales | 09-12-16 | 03-02-17 |
| Test con dispositivo Pressmatic | 16-01-17 | 20-01-17 |
| Búsqueda | 16-01-17 | 17-01-17 |
| Conexión y Ejecución | 18-01-17 | 20-01-17 |
| Documentación | 16-12-16 | 06-03-17 |
| Investigación del tema | 16-12-16 | 23-12-16 |
| Observación de otros proyectos | 30-12-16 | 06-01-17 |
| Elaboración de la memoria | 27-01-17 | 03-02-17 |
| Preparación de la presentación | 17-02-17 | 06-03-17 |

Tabla 22: Organización del trabajo



En esta tabla se muestra, de forma detallada, la organización que se ha llevado a cabo para la realización de este trabajo, desde el aprendizaje del lenguaje y la búsqueda de información hasta la preparación de la presentación.

El inicio del estudio se dio antes de tener los requisitos necesarios para la ejecución del proyecto, puesto que, uno de los objetivos del mismo es el aprendizaje y la familiarización con el desarrollo de aplicaciones Android, así que se llevó a cabo inicialmente un estudio del lenguaje y una investigación de proyectos y aplicaciones.

También se ha incluido en la planificación el tiempo invertido en preparar la presentación. Aunque la fecha final es superior a la entrega de esta memoria, gracias a la experiencia obtenida a lo largo del desarrollo del grado de estudios cursado, se han supuesto tiempos de elaboración y ensayo para la defensa de este trabajo.

Para una completar la descripción de la planificación, se ha incluido un diagrama de Gantt. Este diagrama es una herramienta gráfica en la que se muestra el tiempo que se ha dedicado a cada una de las tareas de un proyecto e indica de forma clara las relaciones entre las fases y la forma en que se interactúa con ellas (estando en la última fase de desarrollo, sigue siendo necesario entender los puntos de la fase de estudio).

Ahora, se adjunta una imagen (figura 33) con el diagrama de Gantt obtenido de la tabla de planificación de la hoja anterior, perteneciente a este trabajo.

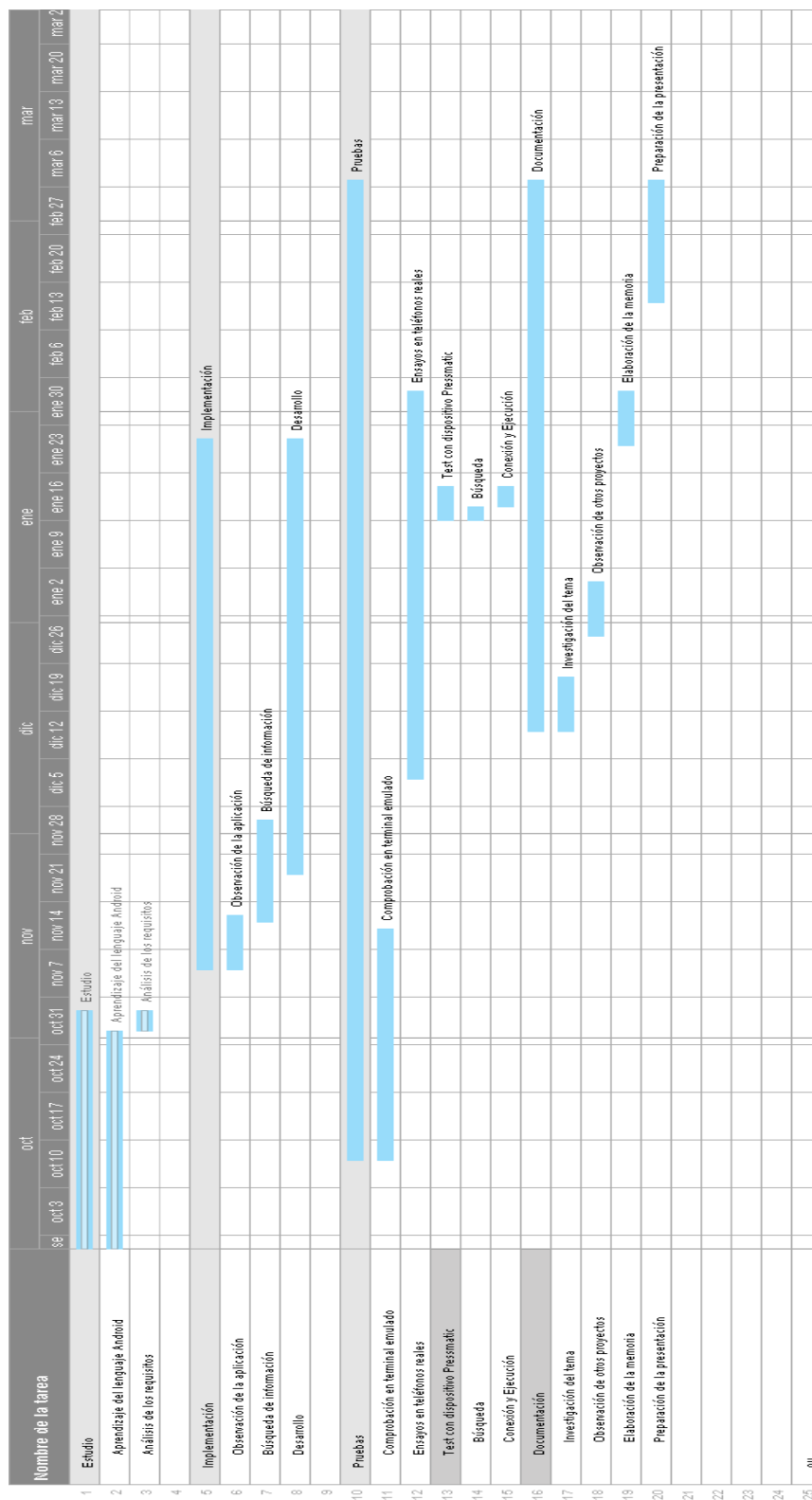


Figura 33: Diagrama de Gantt.



5.4.1 Presupuesto

El presupuesto de este trabajo ha sido calculado teniendo en cuenta los recursos hardware, software y humanos; además del tiempo invertido en su resolución.

- Hardware
 - Ordenador portátil Toshiba Satellite: 512.71€
 - Smartphone Huawei P8 lite: 128€
 - Smartphone Samsung Galaxy Trend: 54.51€
- Software
 - Ubuntu Linux 16.10: 0 €
 - Android Studio + Android SDK: 0 €
 - Microsoft Office 2016: 149.99 €
 - Google Drive: 0 €
 - Adobe Reader: 0 €
- Humanos
 - Coste del desarrollador junior: 16.000 €/año

Los precios mostrados anteriormente no incluyen i.v.a ni en el coste por trabajo humano descuentos por impuestos ni seguridad social.

Por lo tanto, tras cuatro meses de trabajo con 40 días efectivos, el presupuesto quedaría como se muestra en la tabla a continuación, con todos los gastos y retenciones detallados:

**IMPLEMENTACIÓN DE APLICACIÓN ANDROID PARA CONTROL DE DISPOSITIVO ASISTENCIAL
PRESSMATIC**



| RECURSO | COSTE |
|------------------------------|------------------|
| Ordenador portátil | 649€ |
| • Precio | 512.71€ |
| ○ I.V.A (21%) | 136.29€ |
| Smartphone | 238€ |
| • Huawei P8 | 128€ |
| ○ I.V.A | 35.49€ |
| • Samsung Galaxy Trend | 54.51€ |
| ○ I.V.A | 14.49€ |
| Microsoft Office 2016 | 149,99€ |
| ○ Office 2016 | 118.49€ |
| ○ I.V.A | 31.49€ |
| Programador Junior | 1584,4€ * |
| • Sueldo bruto anual | 16.000€ |
| • IRPF | 1.476€ |
| • Seguridad Social | 1.016€ |
| TOTAL | 2622€ |

* (Cantidad por 40 días trabajados. El precio del programador anual sería de 13.507€).

Tabla 23: Precio recursos

El precio del Trabajo fin de grado Implementación de aplicación Android para control de dispositivo asistencial asciende a la cantidad de:

DOS MIL SEISCIENTOS VEINTI DOS EUROS

Firmado

En Madrid, a 27 de febrero de 2017

Javier Ladrón de Guevara Ojalvo



Capítulo 6



Conclusiones y trabajos futuros

Una conclusión de trabajo se da cuando, a partir de sucesos que ocurren de forma provocada o no, se obtiene un entendimiento de la situación (o aumenta el conocimiento respecto a algo).

También, cuando se realiza un trabajo de investigación o en una discusión sobre un tema en particular es necesario terminar con una conclusión, que se basa en un argumento o en una afirmación que sintetiza el trabajo realizado.

En estas conclusiones, se cogen las ideas principales expuestas a lo largo del trabajo y se resume lo investigado, explicando, de forma personal, la razón por la cual se han obtenido esos resultados y cuáles eran los objetivos principales.

6.1 Conclusiones generales

Los objetivos de este proyecto eran la familiarización con el proceso de construcción de una aplicación real Android (planificación, desarrollo, búsqueda de información...) junto con el lenguaje Android, el entorno de programación Android Studio y las pautas de organización aprendidas a lo largo del grado cursado.

Este trabajo ha sido el nexo para entender las diferencias entre la vida universitaria y la laboral, demostrando la utilidad de valores como la constancia, el conocimiento y el uso de los recursos para llevar a cabo las propuestas que se requerían.

Además, el trabajo de investigación realizado ha permitido la profundización y mejora de los conocimientos de las materias implicadas, desde la culturización en cuanto a dispositivos asistenciales hasta el aprendizaje de la arquitectura de cada uno de los sistemas presentes.



6.2 Conclusiones referentes a los objetivos

En este trabajo se pretendía desarrollar una aplicación con tres puntos principales: la capacidad de ser controlada de diferentes formas, la capacidad de cambiar el idioma de la interfaz y nuevas funcionalidades (como la implementación del pulsado continuado de un botón) y la implantación de una interfaz más flexible que la anterior.

A continuación, se detallan estos puntos, exigencias para la resolución de este proyecto:

- Aplicación multilinguaje: capaz de mostrar una interfaz en español e inglés cuando el usuario lo deseara.
- Aplicación multimodal: con una herramienta para poder utilizarla mediante el habla para minimizar el control táctil.
- Diseño dinámico y atractivo para el usuario y desarrollo de nuevas funcionalidades: haciendo la experiencia de ejecución más sencilla y atrayente.

Estos atributos son imprescindibles en las aplicaciones comercializadas hoy en día, pues permiten una experiencia capaz de captar una gran parte de usuarios (desde la variedad en el lenguaje hasta la flexibilidad a la hora de utilizarlas).

Así, el implementar en esta aplicación dichos atributos permite su inclusión en las exigencias actuales y una mejora tanto de su arquitectura como de su comercialización (en el caso de desearlo). Sin olvidar que, las propias actualizaciones del sistema Android permitirán la mejora de las funcionalidades antes descritas, como un avance en el sistema de reconocimiento de voz, en el desarrollo de interfaces, etcétera.



6.3 Trabajos futuros

La actualización, depuración de errores y mejora en la calidad y funcionalidad del código implementado es tradición indispensable en el mundo de la programación, especialmente en el lenguaje Android y la ideología de prosperar de Google.

Los próximos desarrollos podrían basarse en la mejora y aumento de las características que han sido incluidas en la aplicación en este proyecto, dado que se han encontrado errores y hándicaps que han alejado su implementación de la versión ideal propuesta:

- Mejoras en el reconocimiento por voz: progreso del sistema mejorando el reconocimiento en inglés y español, también permitir interpretar frases para el movimiento libre por las pantallas de la aplicación y una funcionalidad para corregir al usuario qué acción tomar dependiendo de la actividad en la que se encuentre.
- Incrementar la calidad del diseño: a través del uso de programas especializados en la calidad de las imágenes, para ajustarlo a diferentes tamaños de pantalla y versiones.
- Perfeccionar el control del estado de la conexión: tanto cuando el dispositivo está conectado como cuando pierde o se desconecta de Pressmatic.

El manejo de la conexión ha sido el principal inconveniente en el desarrollo del proyecto, dado que se utilizan 3 hilos para gestionar la conexión:

1. Escucha las conexiones entrantes
2. Conecta con un dispositivo
3. Realiza la transmisión de datos una vez conectado, pero sólo envían mensajes cada vez que hay un cambio en el estado

Esto dificulta mantener un control continuo del estado de la conexión, ya que al tener la aplicación varias pantallas por las que puede desplazarse el usuario, es



necesario el uso de una variable global que cambie de valor con cada cambio en el estado de la conexión y que lo mantenga al cambiar de actividad.

Una aplicación multilenguaje tanto para el control por voz como táctil era uno de los objetivos del proyecto. En desarrollos futuros se tiene que incluir el reconocimiento completo en ambos idiomas (español e inglés) independientemente del idioma de la aplicación o el teléfono. Ya que la herramienta de reconocimiento utilizada, *ok Google*, utiliza el idioma de defecto del terminal y no de la aplicación, lo que lleva a mal entendidos a la hora de controlar el dispositivo mediante el control por voz.

Aunque esté al margen de los objetivos y metas actuales de las aplicaciones que se desarrollan en este momento, el avance de las tecnologías en la actualidad permite pensar que, para el perfeccionamiento de las aplicaciones asistenciales, es decir, de ayuda a personas con discapacidades, sería útil la incorporación de los hologramas y su vinculación con la tecnología de reconocimiento de voz, sin olvidar los dispositivos que permiten establecer una relación entre las señales del cerebro humano y las órdenes que el individuo está tomando.

Como se pretende el uso de métodos no invasivos y, además, atractivos para generar una experiencia entretenida, los hologramas pueden ser un siguiente paso para la mejora de las aplicaciones de control para dispositivos asistenciales.

Esta unión de tecnologías abarcaría gran parte de las carencias que pueda tener el individuo y facilitaría y mejoraría en su totalidad el uso de estas aplicaciones.



Figura 34: Logo Pressmatic



Referencias

- [1]. Rechac Ángela: Rediseño de la app de control remote de pressmatic. Universidad Carlos III de Madrid, Madrid. 2015.

- [2]. E.D. Oña, A. Jardon, C. Balaguer, A. Martínez, P. Sánchez-Herrera, J.C. Miangolarra. A pilot study on the design consideration and user impressions of an assistive affordable device. RoboCity16 Open Conference on Future Trends in Robotics. RoboCity16 Open Conference on Future Trends in Robotics, ISBN: 978-84-608-8452-1, pages: 19 - 28. 2016-05-26, Madrid, Spain. 2016.

- [3]. Assistireland: Apps for People with disabilities and older people.
[http://www.assistireland.ie/eng/Information/Information Sheets/Apps for People with Disabilities and Older People.html](http://www.assistireland.ie/eng/Information/Information_Sheets/Apps_for_People_with_Disabilities_and_Older_People.html)

- [4]. Cárdenas Xavier: Uso de la robótica en la medicina Universidad Politécnica Salesiana.
[https://upload.wikimedia.org/wikipedia/commons/5/50/USO DE LA ROBOTICA EN LA MEDICINA.pdf](https://upload.wikimedia.org/wikipedia/commons/5/50/USO_DE_LA_ROBOTICA_EN_LA_MEDICINA.pdf)



- [5]. Efe: Futuro. España, puntera en robótica asistencial dirigida a ancianos y enfermos. 2016.
<http://www.efefuturo.com/noticia/espana-puntera-en-robotica-asistencial-dirigida-a-ancianos-y-enfermos/>
- [6]. Mulhern Nicholas, McCaffrey Neil, Beretta Nicholas, Chabot PhD Eugene, Ying Sun PhD: Designing Android Applications using Voice Controlled Commands. University of Rhode Island Department of Electrical, Computer and Biomedical Engineering. 2013.
- [7]. Shraddha Uddhav khadilkar, Narendra Wagdarikar: Android phone controlled Voice, Gesture and Touch screen operated Smart Wheelchair. Dnyanganga College of Engineering and Research Pune, India. 2015.
- [8]. Agencia Estatal. Boletín oficial del Estado: Legislación consolidada.
<https://www.boe.es/buscar/act.php?id=BOE-A-2008-979>
- [9]. Invarato Menéndez Ramón: Android 100%. España. 2014-09-24.
<https://jarroba.com/libro-android-100-gratis/>



- [10].** Terra, ecología práctica: Impacto ambiental de móviles y microelectrónica.
<http://www.terra.org/categorias/articulos/impacto-ambiental-de-moviles-y-microelectronica>
- [11].** ParqueUc3m: Laboratorio de robótica asistencial. Madrid, España. 2012
<https://www.youtube.com/watch?v=6SFttXWI3kQ>
- [12].** Información y consulta: Trabajos fin de grado. Universidad Carlos III de Madrid. Madrid, España. 2016
<http://e-archivo.uc3m.es/>
- [13].** Daniel's Castañeda: Historia la ingeniería rehabilitación Academia Edu.
[http://www.academia.edu/9445420/Historia la ingenier%C3%ADa rehabilitaci%C3%B3n](http://www.academia.edu/9445420/Historia_la_ingenier%C3%ADa_rehabilitaci%C3%B3n)
- [14].** RoboticsLab Uc3m: Pressmatic. Universidad Carlos III de Madrid. Madrid, España. 2016
<http://roboticslab.uc3m.es/roboticslab/project/pressmatic>
- [15].** AppStonic: ¿Que es Android? ¿De qué está formado? Conoce su estructura. 2013
<http://www.appstonic.com/que-es-android/>



[16]. Desarrolladores Android: Introducción a Android.

<https://developer.android.com/guide/index.html>

[17]. Desarrolladores Google: Android

<https://developer.android.com/index.html>

[18]. Comunidad Stack Overflow: Consultas Android.

<http://stackoverflow.com/>

[19]. García Daniel: Bluetooth en Android. 2013

<https://danielggarcia.wordpress.com/2013/10/19/bluetooth-i-activando-y-desactivando-el-bluetooth-en-android/>

[20]. Profesor Molina: Historia de la Robótica.

<http://www.profesormolina.com.ar/tecnologia/robotica/historia.htm>

[21]. Fundación diseño para todos.

<http://designforall.org/design.php?Setlang=es>

[22]. BERMEJO, J. C. 2013

www.josecarlosbermejo.es/articulos/el-compromiso-de-la-humanización-en-las-instituciones-sociosanitarias



Capítulo 7

Anexos

7.1 Manual de instalación

El IDE de programación elegido ha sido Android Studio (última versión disponible en la web oficial [Android Studio](https://developer.android.com/studio)), pues es el acogido por los desarrolladores de Android y el que posee actualmente más soporte online. En la página web de instalación indican paso a paso cómo configurar el entorno de programación (versión Java, paquete Android SDK...) según el sistema operativo elegido (Linux, Windows...).

La versión de Android debe ser 6.0 o superior, y la mínima la 3.0 (al menos versión SDK 15).

File -> Settings -> Android SDK -> Instalación de los paquetes deseados

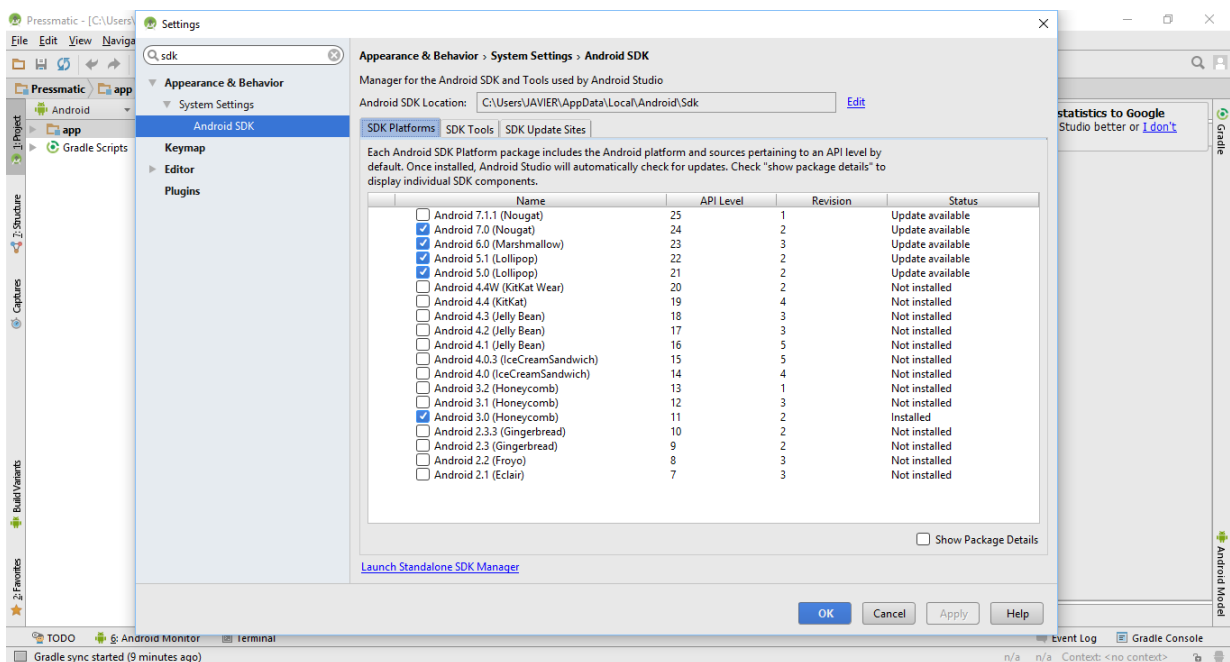


Figura 35: Manual de instalación.



Para ejecutar el programa se abre desde el menú:

File-> Open -> Elegir el logo del software Android Studio con el nombre de Pressmatic.

- Se conecta el teléfono con el ordenador mediante un cable usb y así implementar la aplicación en el teléfono.
- Para ejecutar la aplicación se pulsa el triángulo verde en las opciones del menú superior del programa, como emulador se elige el nombre del teléfono.

Es necesario habilitar la opción de depuración en el teléfono para que permita utilizarlo como emulador de aplicaciones de Android Studio, aquí dejo un tutorial con los pasos:

- Coge el móvil y activa la depuración USB. *¿Cómo activo la depuración USB?* Desde las Opciones de Desarrollo. *¿Cómo activo las opciones de desarrollo?* > Coge tu smartphone > **Ajustes > Información del teléfono > Pulsa varias veces Número de compilación**. Cuando hagas esto, habrás activado las Opciones de desarrollo, que encontrarás en el menú anterior, en la ventana de Ajustes. Ahora ya puedes entrar y activar “**Depuración por USB**”.

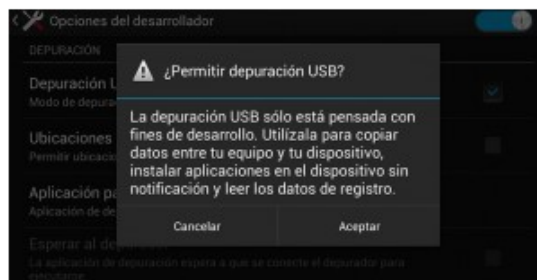


Figura 36: Manual de instalación 2.



7.2 Manual de usuario

La aplicación de Pressmatic comienza con las instrucciones de uso, en las que brevemente se explica su funcionamiento mediante imágenes y una pequeña descripción. Además, aparece un mensaje de alerta para activar el Bluetooth del teléfono y conectar con la red del dispositivo para utilizar la aplicación tras la segunda pantalla de instrucciones.

También es posible cambiar el idioma de la aplicación y acudir a las instrucciones desde cualquier pantalla cuando sea necesario utilizando los iconos de rueda (para cambiar lenguaje español/inglés) e interrogación (para acceder al tutorial).

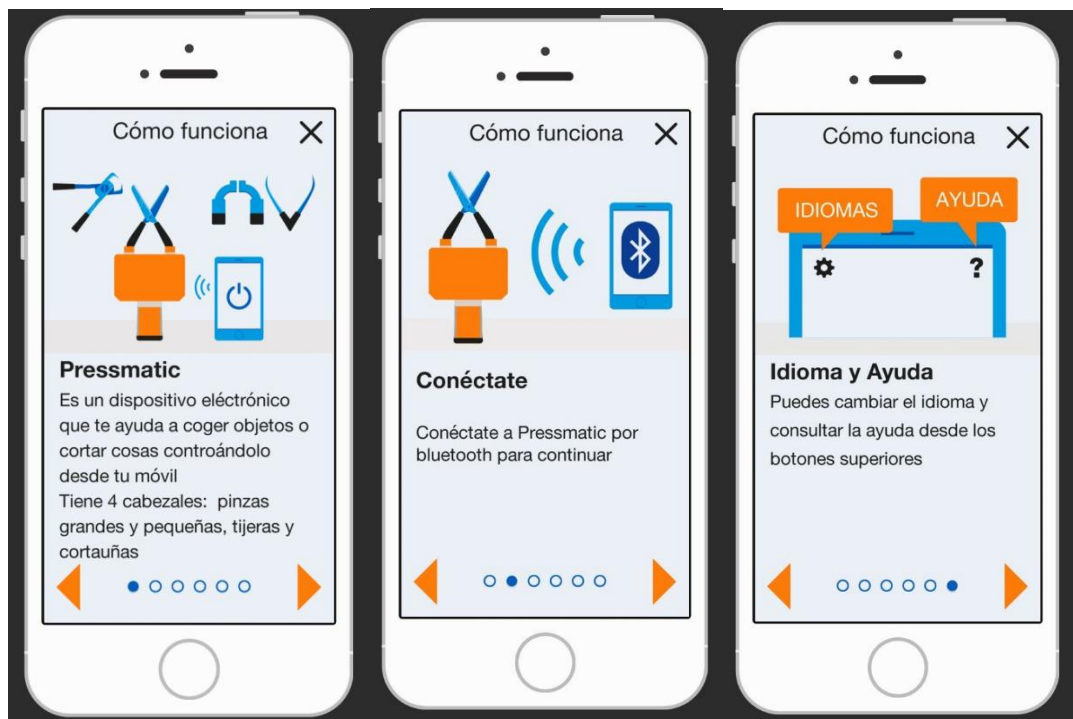


Figura 37: Manual de instrucciones 1.



Al pulsar el botón “X” para salir de las instrucciones aparece la pantalla de conexión con el dispositivo: si el bluetooth del teléfono está activado, indicará que es necesario conectar con Pressmatic; sino, será necesario habilitar el bluetooth del teléfono para continuar con la aplicación (si se elige la opción no y se intenta conectar con el dispositivo directamente, volverá a aparecer la misma pantalla).



Figura 38: Manual de instrucciones 2: Bluetooth. Izquierda: habilitar. Derecha: conectar.

Después de conectar con el dispositivo y recibir la confirmación por parte de Pressmatic, aparecerá un nuevo menú de selección de herramienta automáticamente en la aplicación. El usuario podrá navegar de izquierda a derecha para elegir uno de los utensilios (también podrá cambiar el idioma y consultar las instrucciones mediante los botones de ajustes e interrogación, respectivamente).

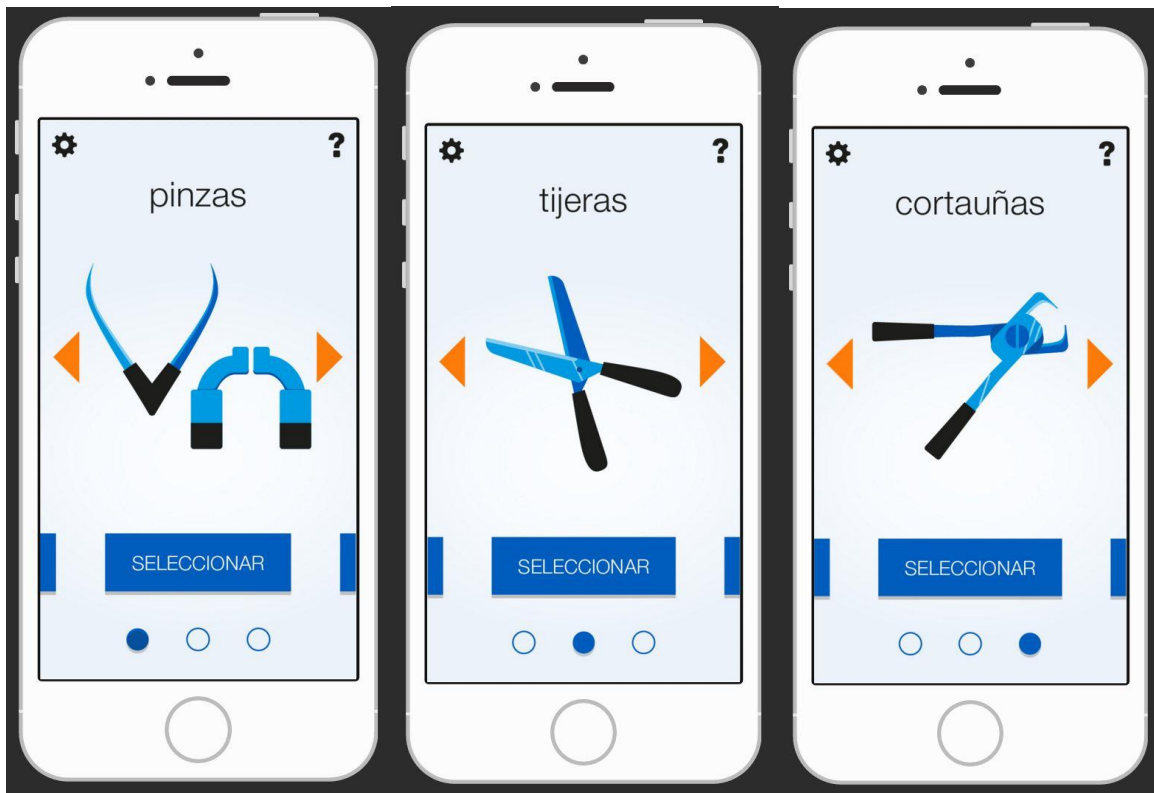


Figura 39: Manual de instrucciones 3: Herramientas disponibles.

A continuación, se indica que, tras elegir una de las herramientas disponibles, es posible ejecutar acciones con pulsar el botón indicado según el aparato escogido (on/off, cortar, abrir/cerrar). Además de tener la posibilidad de modificar la velocidad de acción con cada uno de ellos.

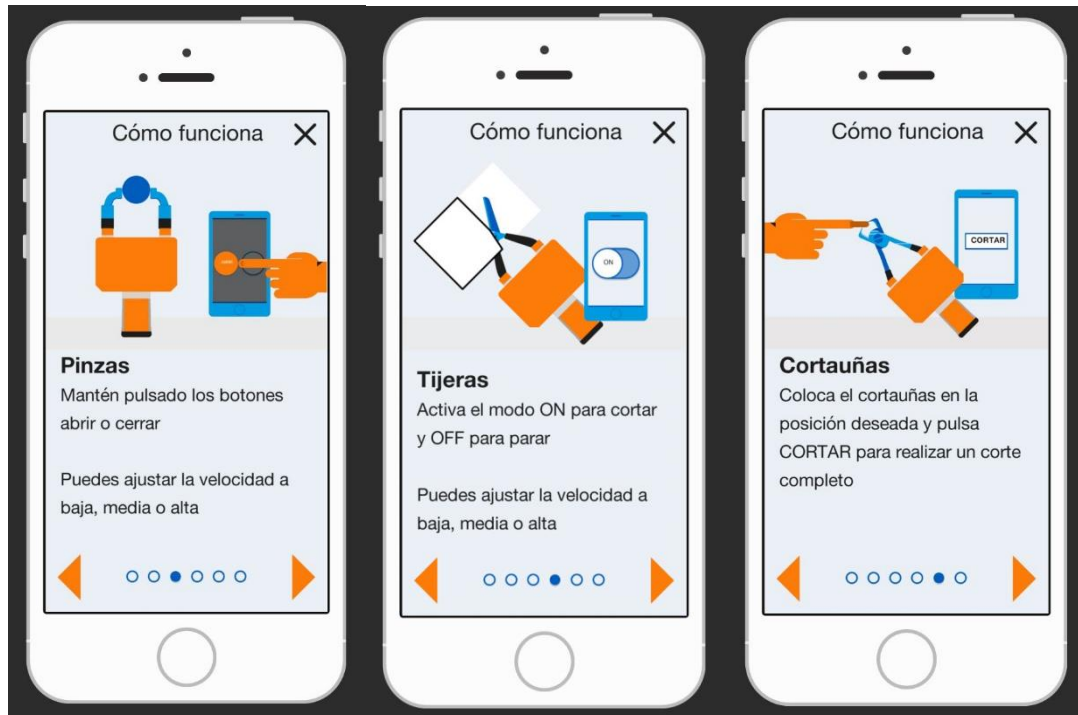


Figura 40: Manual de instrucciones 4.

La opción de control por voz está disponible en todas las pantallas y se utiliza pulsando una vez el icono del micrófono ("habla ahora" de Google) presente en todas las actividades (en mitad de la pantalla) y pronunciando una palabra (en español o inglés) para ordenar a la aplicación la ejecución deseada (izquierda, derecha, cerrar, conectar, cortar...). La aplicación sólo reconoce una palabra por ejecución del reconocedor, no admite frases ni órdenes complejas.



Figura 41: Manual de instrucciones 5.

Para utilizar el reconocimiento de voz únicamente es necesario tener disponible conexión a internet o activada la opción de “operar sin conexión” en las opciones del dispositivo. Para ello se deben seguir los siguientes pasos:

- Desde la aplicación de Google: Ajustes -> Voz -> Reconocimiento de voz sin conexión.
- Desde el terminal: Ajustes -> Idioma e introducción de texto -> Dictado por voz de Google -> Reconocimiento de voz sin conexión.

IMPLEMENTACIÓN DE APLICACIÓN ANDROID PARA CONTROL DE DISPOSITIVO ASISTENCIAL
PRESSMATIC

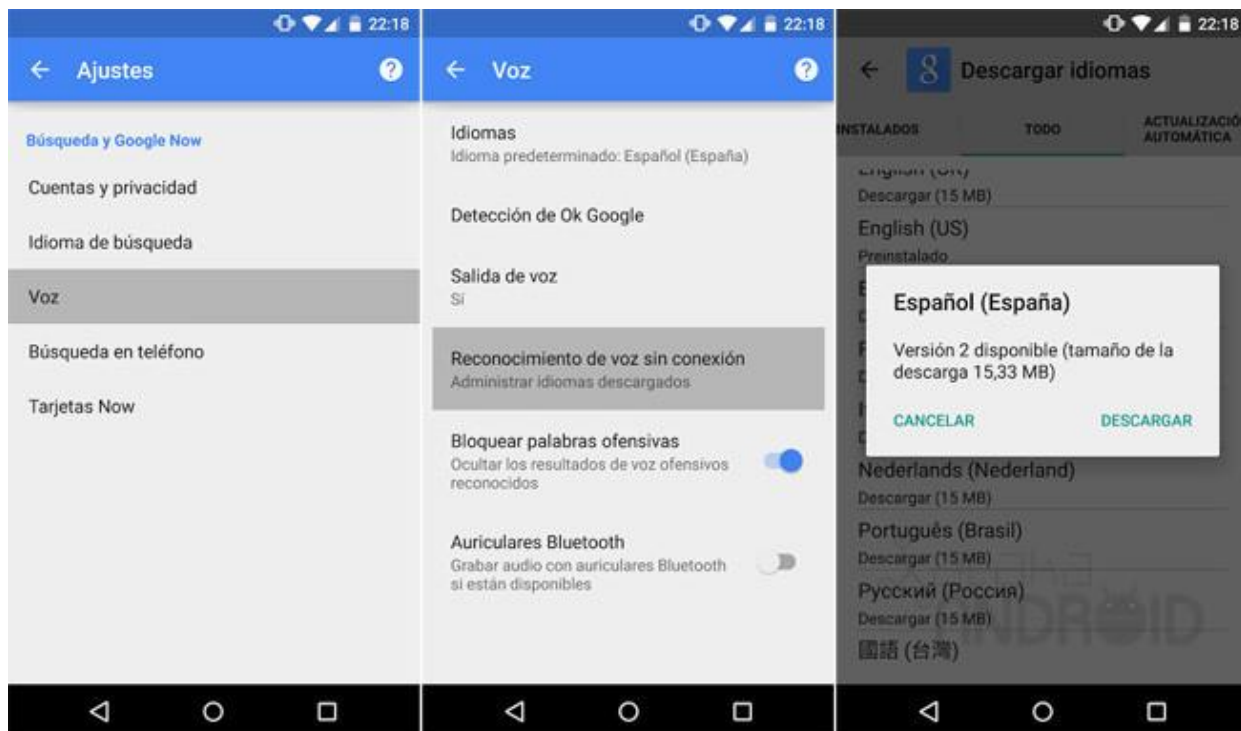


Figura 42: Manual de instrucciones 6.